



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**AKUSTICKÁ DETEKCE POZICE ŘEČNÍKA POMOCÍ MI-
KROFONNÍHO POLE A JEJÍ VIZUALIZACE V PROSTORU**

ACOUSTIC DETECTION OF SPEAKER POSITION USING MICROPHONE ARRAY AND VISUALI-
SATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ZDENĚK PELZ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE, Ph.D.

BRNO 2019

Abstrakt

Tato práce řeší problematiku lokalizace řečníka v prostoru pomocí mikrofonního pole. Cílem této práce je nastudovat algoritmy pro lokalizaci řečníka v prostoru a experimenty s těmito algoritmy. Problém určování TDOA jsem řešil pomocí korelace signálů a pozici jsem počítal pomocí hyperbolické metody. Výsledné pole a algoritmy jsou schopné lokalizovat řečníka s jistou odchylkou. Výsledky práce umožňují udělat si představu o přesnosti lokalizace řečníka pomocí mikrofonního pole a ARM sestavy s omezeným výkonem. Výpočet pozice za použití mikrofonního pole dosahovalo řádové přesnosti jednotek decimetrů ovšem tato přesnost se lišila se vzdáleností od mikrofonního pole.

Abstract

This thesis explores problematics of speaker localization using microphone array. Aim of this thesis is implementation of algorithms for speaker localization and experiments with those algorithms. Calculation of TDOA was done using cross-correlation and hyperbolic method was used to calculate position estimation. Finished microphone array is able to locate speaker within certain variance. Results of this thesis allow reader to make assumptions regarding accuracy of localisation using microphone array and ARM kit with limited performance. Precision of position estimation using microphone array reached several decimeters, but this precision is dependent on distance from microphone array.

Klíčová slova

mikrofonní pole, TDOA, lokalizace pomocí zvuku, hyperbolická metoda, křížová korelace

Keywords

microphone array, TDOA, acoustic localization, hyperbolic method, cross-correlation

Citace

PELZ, Zdeněk. *Akustická detekce pozice řečníka pomocí mikrofonního pole a její vizualizace v prostoru*. Brno, 2019. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Igor Szöke, Ph.D.

Akustická detekce pozice řečníka pomocí mikrofonního pole a její vizualizace v prostoru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szöke, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Zdeněk Pelz
21. května 2019

Poděkování

Rád bych poděkoval panu Ing. Igorovi Szökemu, Ph.D. za vedení této práce a za jeho nápady a připomínky na konzultacích.

Obsah

| | | |
|----------|--------------------------------------------------------------------|-----------|
| 1 | Úvod | 3 |
| 1.1 | Cíl práce | 3 |
| 2 | Teoretická část práce | 5 |
| 2.1 | Mikrofonní pole | 5 |
| 2.2 | Zvuk | 7 |
| 2.3 | Filtrování signálu | 8 |
| 2.4 | Metody lokalizace | 9 |
| 2.5 | Time Difference of Arrival - TDOA | 9 |
| 2.6 | Time of Arrival - TOA | 10 |
| 2.7 | Použití TOA a TDOA | 12 |
| 2.8 | Beamforming | 12 |
| 2.9 | Křížová korelace | 13 |
| 2.10 | Hyperbolická metoda určení pozice | 17 |
| 2.11 | Optimalizovaná hyperbolická metoda pro určení polohy | 19 |
| 2.12 | Metoda využívající síly signálu | 20 |
| 3 | Pokusné mikrofonního pole | 22 |
| 3.1 | Použitý hardware | 23 |
| 3.2 | Popis rozdělení úloh mezi PC a mikrokontrolér | 25 |
| 3.3 | Křížová korelace | 25 |
| 4 | Detaily implementace prototypu | 27 |
| 4.1 | Data z mikrofonního pole | 27 |
| 4.2 | Detekce zvuku | 29 |
| 4.3 | Křížová korelace | 29 |
| 4.4 | Výpočet polohy | 29 |
| 4.5 | Časově hyperbolická metoda lokalizace | 29 |
| 4.6 | Pomocné grafické rozhraní | 30 |
| 5 | Testování | 32 |
| 5.1 | Přesnost korelace | 32 |
| 5.2 | Přesnost algoritmu pro lokalizaci v ideálních podmínkách | 33 |
| 5.3 | Lokalizace stacionárního cíle | 34 |
| 5.4 | Sledování pohyblivého cíle | 35 |
| 6 | Sestava ARM/SHARC | 38 |
| 6.1 | Procesor SHARC | 38 |

| | | |
|----------|------------------------------------------------------------|-----------|
| 6.2 | Použité mikrofony | 39 |
| 6.3 | Práce se sestavou | 39 |
| 6.4 | Překladač | 39 |
| 6.5 | ALSA knihovna | 40 |
| 6.6 | Struktura mikrofonního pole | 40 |
| 6.7 | Struktura programu | 41 |
| 7 | Výsledky lokalizace zdroje zvuku pomocí ARM sestavy | 43 |
| 7.1 | Lokalizace statického zdroje zvuku | 43 |
| 7.2 | Lokalizace pohyblivého zdroje zvuku | 45 |
| 7.3 | Změny parametrů pro lokalizaci | 45 |
| 8 | Závěr | 47 |
| 8.1 | Přesný výpočet pozice | 47 |
| 8.2 | Možnosti dalšího vývoje | 48 |
| | Literatura | 49 |
| | A Plakát | 50 |
| | B Obsah paměťového média | 51 |

Kapitola 1

Úvod

V této práci se řeší problematika lokalizace zdroje zvuku pomocí mikrofonního pole, způsoby pořizování nahrávek z prostoru, jejich zpracování a určení pozice zdroje.

Je zde rozebrána struktura mikrofonních polí, která může mít vliv na výkon a přesnost mikrofonních polí. Nevhodně uspořádané pole může dosahovat horších výsledků. Dále je zde popsán způsob zpracování signálu, který mikrofonní pole nahrálo. Pokud je kvalitní nahrávka ale nepřesné algoritmy, celý výpočet pozice zdroje zvuku může být nepřesný.

Problematiku výpočtu pozice podle zvuku zachyceného mikrofonním polem lze rozdělit na dvě části. Pokud je přijatý signál před zpracováním vhodným způsobem upraven, předzpracování může značně usnadnit a urychlit práci s tímto signálem. Úprava signálu proto tvoří významnou část procesu lokalizace zdroje zvuku. Po dokončení všech potřebných úprav signálu přichází na řadu samotný výpočet polohy zdroje zvuku. Zde je zapotřebí zvolit vhodný algoritmus pro použitý typ mikrofonního pole. Zvolení špatného algoritmu může výpočet polohy velmi ztížit nebo dokonce znemožnit. Výpočet polohy je druhá a nejdůležitější částí celého procesu lokalizace.

Problematika mikrofonních polí je v dnešní době populární a značně zkoumané téma. Využití mikrofonních polí sahá od jednoduchých použití při zlepšování kvality pořízených nahrávek až po komplexní úlohy související s lokalizací cíle v prostoru [12]. Lokalizací cíle v prostoru rozumíme úlohu, kdy pomocí mikrofonů nahrajeme zvukovou stopu nějakého zdroje zvuku a pomocí algoritmů vypočítáme pozici řečníka vůči mikrofonnímu poli popř. jakéhokoliv bodu v prostoru.

V následujících kapitolách je teoretický i praktický popis vytvořeného mikrofonního pole, výčet použitých algoritmů a postup při jejich vytváření a testování.

1.1 Cíl práce

Cílem této práce je nastudovat problematiku lokalizace zdroje zvuku pomocí mikrofonního pole, provést experimenty s polem mikrofonů a implementovat algoritmy, pomocí kterých bude možno detekovat a lokalizovat zdroj zvuku za použití tohoto pole.

Pole nebude schopné detekovat více zdrojů zvuku současně. Pokud bude v místnosti více zdrojů zvuku současně, výpočet pozice zdroje zvuku bude značně nepřesný. To je způsobeno zejména tím, že použité algoritmy, pro výpočet pozice zdroje zvuku, nejsou uzpůsobeny k detekci více zdrojů.

Tato práce může dále sloužit jako příručka popř. návod pro experimenty s mikrofonními poli, jejich strukturou a algoritmy, které s nimi budou pracovat

1.1.1 Použití vlastního prototypu mikrofonního pole

Lokalizace zdroje zvuku měla probíhat za použití ARM sestavy, která byla zapůjčena školou. Tato sestava ovšem měla být k dispozici až na konci zimního semestru. Pro pokusy s lokalizací za použití mikrofonního pole jsem se tedy rozhodl sestavit si vlastní mikrofonní pole za použití desky Blue pill kompatibilní s vývojovým prostředím Arduino a pěti mikrofونů a začít experimenty s tímto mikrofonním polem.

Jako další možnost se nabízely experimenty s nahrávkami zvuku jedním mikrofونem, a takto nahraný signál nakopírovat a dále časově zpozdít, aby se jevíly jako nahrávky z více mikrofونů a na těchto nahrávkách testovat veškeré algoritmy. Pro větší autenticitu jsem se ovšem rozhodl sestavit prototyp mikrofonního pole, aby byly algoritmy lépe připravené pro použití s reálným polem.

ARM sestavu jsem obdržel ke konci druhého semestru a proto většina vývoje algoritmů pro lokalizaci probíhala za využití tohoto prototypu.

Po obdržení ARM sestavy jsem začal přepisovat algoritmy pro fungování na ARM sestavě a začal testovat funkčnost algoritmů na této sestavě.

Kapitola 2

Teoretická část práce

V této části práce jsou popsána teoretická řešení problémů spojených s lokalizací zdroje signálu.

Lokalizace zdroje signálu je velmi obtížný proces z několika důvodů. Za prvé je nutné rozmístit mikrofony takovým způsobem, aby pokrývali co největší prostor. Pokud nejsme schopni zachytit signál, který vydává zdroj, nejsme schopni jej pomocí tohoto signálu ani lokalizovat. Dále je potřeba korektně zpracovat přijatý signál. K dispozici máme několik signálů zachycených rozmístěnými senzory. Tyto signály nejspíše obsahují šum a jiné nežádoucí prvky jako např. ozvěna, signály pocházející z jiných zdrojů atd. Proto je potřeba signál upravit filtrací nebo jinými způsoby. Po zpracování přichází na řadu samotná lokalizace zdroje signálu. Lokalizovat zdroj je možné několika způsoby, ovšem v reálných výpočtech při lokalizaci dochází k nepřesnostem, které je zapotřebí řešit. Tyto nepřesnosti mohou být způsobené několika příčinami. Na rozdíl od použitelnosti algoritmů v teoreticky ideálním prostředí může při reálném použití nastat mnoho problémů. Použitelností je zde myšlena přesnost algoritmu, jeho rychlost a nebo paměťové nároky. Jedním z těchto problémů může být např. rychlost algoritmů. Algoritmy, které mají dobré výsledky v ideálních podmínkách nemusí být tou správnou volbou při použití v reálných podmínkách. Mohou být sice přesné, ovšem pro použití v reálných aplikacích mohou být příliš pomalé. Při reálném použití dochází ke kompromisu mezi přesností a rychlostí algoritmů. Volba a technické provedení těchto algoritmů záleží z velké části na zaměření aplikace.

2.1 Mikrofonní pole

Mikrofonní pole je zařízení sestávající se z množství mikrofonů rozmístěných takovým způsobem, aby bylo co nejlépe možné pořizovat nahrávky z celého prostoru, který toto mikrofonní pole má za úkol pokrývat. Každý mikrofón můžeme chápat jako jeden akustický kanál, se kterým je možné nadále pracovat. Hlavním úkolem zpracování signálů z mikrofonních polí může být několik. Pomocí mikrofonního pole lze lokalizovat zdroj zvuku. Dále je možné použít mikrofonní pole k pořízení nahrávky vyšší kvality, než jaké se dá dosáhnout při použití jednoho mikrofónu [1]. Této vyšší kvality lze dosáhnout sečtením patřičně časově posunutých signálů z jednotlivých mikrofonů. V teorii by měla energie hledaného signálu být podstatně větší než energie případného šumu, který by se měl navzájem vyrušit.

Rozestavení mikrofonů a celková geometrie mikrofonního pole může také hrát významnou roli, ovšem záleží na zaměření, aplikaci nebo použitých algoritmech [1]. Při řešení lokalizace pomocí mikrofonního pole musí být struktura mikrofonního pole předem známá

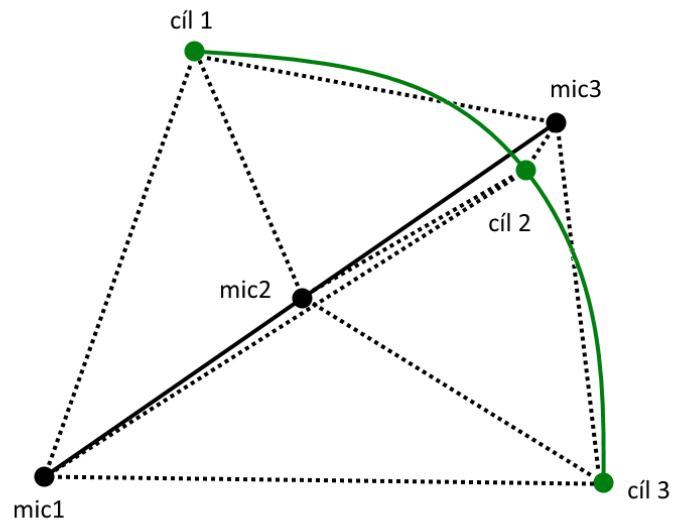
abychom mohli přesně lokalizovat zdroj zvuku. V některých případech můžou např. pravidelné rozestupy, mezi mikrofony pole, problém lokalizace výrazně usnadnit.

2.1.1 Geometrie mikrofonního pole

Z hlediska rozestavení mikrofونů v poli lze mikrofonní pole rozdělit na tři různé typy:

- Lineární mikrofonní pole - toto mikrofonní pole je tvořeno mikrofony umístěnými v jedné přímce.
- Rovinné mikrofonní pole - toto mikrofonní pole je tvořeno mikrofony umístěnými v jedné rovině.
- Prostorové mikrofonní pole - toto mikrofonní pole je tvořeno mikrofony libovolně rozmístěnými v prostoru.

Rozestavení mikrofونů v poli také určuje do jaké míry jsme schopni lokalizovat zdroj přijatého signálu. Pokud jsou mikrofony umístěny pouze v jedné přímce, nejsme schopni lokalizovat zdroj signálu v 3D prostoru. Na obrázku 2.1 lze vidět schéma lokalizace zdroje signálu ve 3D prostoru pomocí lineárního pole.



Obrázek 2.1: Problematika lokalizace zdroje signálu v 3D prostoru za použití lineárního pole. Jsou zde vidět 3 cíle, u kterých mikrofonní pole určí stejnou polohu i přesto, že se nachází na jiných souřadnicích.

Z obrázku je patrné že pomocí tohoto pole jsme schopni určit x a y souřadnici zdroje signálu vůči mikrofonnímu poli. Ovšem pokud se bude zdroj pohybovat po trajektorii s konstantní vzdáleností od přímky, na které jsou umístěny mikrofony pole, pomocí lineárního pole tento pohyb nelze rozeznat. Pro lokalizaci v prostoru je tedy potřeba umístit mikrofony tak, aby se pomocí jejich pozic dala jednoznačně definovat rovina tzn. mikrofony musí tvořit alespoň rovinné mikrofonní pole.

2.1.2 Rozestupy mezi mikrofony

Vzdálenost mezi mikrofony záleží na účelu, pro který je mikrofonní pole navrženo. Problém ovšem nastává pokud je vzdálenost mezi mikrofony, d , větší než velikost časového úseku, t , za kterou měřený signál urazí vzdálenost $\frac{c}{t}$, kde c je rychlost signálu. Pokud máme délku časového úseku měřeného signálu 10 ms, zvuk za tuto časovou dobu urazí 3.43 m. Pokud budeme uvažovat, že zdroj zvuku leží na přímce tvořené dvěma mikrofony a tyto mikrofony budou od sebe vzdálené více než 3.43 m a budou nahrávat pouze 10 ms, dojde k tomu, že tyto nahrávky budou obsahovat zvukové stopy, které se nebudou překrývat, což lze považovat např. při výpočtu korelace těchto signálů za problém.

2.1.3 Použití mikrofonních polí

Mikrofonní pole našla v posledních letech hned několik využití, ovšem stále přetrvávají dvě hlavní použití. Mikrofonní pole se používají zejména pro lokalizaci zdroje zvuku a pořizování kvalitnějších nahrávek, než co je možné s jedním mikrofonem [12].

2.2 Zvuk

Zvuk je energie, která je tvořena vibracemi. Pokud nějaké těleso vibruje, vytváří to pohyb mezi částicemi nosiče. Tyto vibrace se dále šíří nosičem ve formě tlakové vlny¹.

2.2.1 Šíření zvuku v prostoru

Zvuk se šíří pouze ve hmotném prostředí. Na tomto prostředí je také závislá jeho rychlost. Zvuk se v prostředí šíří od zdroje všemi směry a je možné toto šíření popsat vlnoplochami. Vlnoplocha je množina bodů v prostoru, které kmitají se stejnou fází. Vlnoplochy lze rozdělit do dvou skupin:

- Kulové vlnoplochy - vznikají v blízkosti menších zdrojů vlnění, mají tvar koule
- Rovinné vlnoplochy - vznikají pokud je zdroj značně rozměrný, ovšem za rovinné vlnoplochy lze považovat i kulové vlnoplochy, které urazily velkou vzdálenost od zdroje, mají tvar roviny

Šíření zvuku v prostoru podléhá Huygensově principu.

Huygensův princip

Každý bod vlnoplochy, do něhož dospěje vlnění v určitém okamžiku, se stává novým zdrojem elementárního vlnění. Vlnoplocha je množina bodů, které kmitají se stejnou fází. Vnější obalová plocha všech elementárních vlnoploch tvoří výslednou vlnoplochu v dalším časovém okamžiku².

2.2.2 Změny rychlosti zvuku

Rychlost šíření zvuku ve vzduch je obecně známá jako $v = 343\text{ms}^{-1}$. Ovšem rychlost zvuku je proměnlivá. Rychlost zvuku ovlivňuje řada faktorů, zejména teplota prostředí a

¹<https://www.encyclopedia.com/science-and-technology/physics/physics/sound>

²https://www.wikiskripta.eu/w/Huygensův_princip

atmosférický tlak. Další faktory, které mohou ovlivnit rychlost šíření zvuku jsou vlhkost vzduchu a koncentrace CO_2 ve vzduchu [4].

Podle Boylova zákona je rychlost zvuku, v , v ideálním plynném prostředí dána vztahem popsaným v rovnici 2.1:

$$v = \sqrt{\frac{\gamma RT}{M}} \quad (2.1)$$

kde γ je poměr tepelné kapacity za konstantního tlaku, R je plynová konstanta, T značí absolutní teplotu a M je molární hmotnost. Z této rovnice je patrné, že teplota má jistý vliv na rychlost zvuku [6].

Předpokladem pro přesné fungování odhadu polohy je tedy aby mikrofonní pole bylo instalováno v místnosti. Při uvažování této skutečnosti můžeme výpočet rychlosti zvuku zanedbat a při uvažování pokojové teploty, za kterou se obecně považuje teplota v rozsahu $18 - 24^\circ C$, lze považovat rychlost zvuku za konstantní. Jako rychlost zvuku při pokojové teplotě lze uvažovat konstantu $343.6ms^{-1}$.

Pokud se ovšem na tuto konstantu nedá spolehnout a je zapotřebí přesnějšího výpočtu této rychlosti, bude potřeba počítat tuto rychlost v reálném čase za pomoci přídavného termometru jako součásti mikrofonního pole.

2.2.3 Šířka frekvenčního pásma řeči

Normální šířka frekvenčního pásma lidské řeči se pohybuje v intervalu od 80 Hz do 3 kHz. Ovšem trénovaný hlas dokáže při zpěvu dosáhnout frekvencí okolo 3,6 kHz [2]. Tyto abnormálně vysoké frekvence ale dále nebudeme brát v úvahu.

Pokud se v signálu budou vyskytovat frekvence mimo toto pásmo, zjevně se jedná o šum, který není schopná vyprodukovat průměrná osoba a je potřeba tyto frekvence odfiltrovat.

2.3 Filtrování signálu

Filtrování signálu je nedílnou součástí při úlohách kde je zapotřebí zpracovávat signál. Signál může být při šíření prostorem různými způsoby deformován či zkreslen. Tím dochází ke snížení kvality signálu a znehodnocení informací, které nese. Pokud je zapotřebí odstranit tyto nechtěné změny signálu lze k tomu použít filtry.

Obecně lze rozdělit použití filtrů do dvou základních kategorií:

- separace více signálů, které byly nějakým způsobem zkombinovány
- obnovení zkresleného signálu

Separace signálů je zapotřebí pokud byl signál kontaminován rušením, šumem nebo jiným signálem. Pokud je původní signál příliš kontaminován a vyžaduje se jeho další použití, je nutné tento signál oddělit od šumu a ostatních deformací [11].

Obnova signálu se používá pokud byl signál nějakým způsobem zkreslen či deformován. Pokud je zapotřebí s tímto signálem dále pracovat a míra deformace je příliš vysoká, je nutné signál upravit pomocí filtrace. Toto zkreslení může být důsledkem mnoha vlivů např. špatným vybavením pro pořizování záznamu [11].

Filtry lze rozdělit do těchto skupin:

- Analogové filtry

- Digitální filtry

K řešení problémů uvedených výše lze použít jak analogové tak i digitální filtry. Analogové filtry mají ovšem oproti digitálním filtrům několik nevýhod. Analogové filtry jsou levné, rychlé a mají velký rozsah pro zpracování jak amplitud, tak frekvencí. Digitální filtry jsou v porovnání s analogovými ovšem mnohonásobně rychlejší. Tato skutečnost představuje velký rozdíl v přístupu k problémům zahrnujícím filtrování [11].

Při zpracovávání zvuku při lokalizaci je žádoucí provést filtraci. Je potřeba odstranit ze signálu nežádoucí šum, který pochází z okolí. Pokud je v signálu přítomen hluk, je potřeba tuto skutečnost řešit. Lidská řeč má ovšem relativně úzké frekvenční pásmo, můžeme si tedy dovolit odfiltrovat veškeré frekvence mimo toto pásmo. Pro řešení tohoto problému lze použít např. **band-pass** filtr, který ořeže všechny frekvence větší nebo menší než předem specifikované prahové hodnoty.

2.4 Metody lokalizace

Metody lokalizace pomocí pole senzorů lze rozdělit do dvou skupin:

- Aktivní lokalizace
- Pasivní lokalizace

Pokud objekt vyzařuje nebo odráží signál, informace o cíli, především o jeho vzdálenosti a celkové poloze, je obsažena v prodlevě, kdy signál dorazí k cíli a odrazí se zpět k senzorům čekajícím na tento signál. Pokud je časový průběh signálu předem známý, je možné srovnáním fázového posunu původního signálu se signálem zpožděným touto cestou zjistit vzdálenost, kterou původní signál urazil. Tento jev je využíván u aktivních lokátorů jako např. radar nebo sonar [10].

Pokud předem neznámý zdroj vyzařuje signál o určité vlnové délce, pomocí senzorů lze měřit pouze časové rozdíly dopadu signálu na pole senzorů. Takový způsob zpracování obdržených dat se nazývá pasivní lokalizace [10].

2.5 Time Difference of Arrival - TDOA

Časový rozdíl návratu signálu (dále jen TDOA) vyjadřuje rozdíl v dopadu fázově posunutého signálu na jednotlivé prvky pole senzorů. Předpokládejme, že toto pole senzorů tvoří mikrofonní pole. Z tohoto časového rozdílu v dopadech signálu lze vypočítat polohu zdroje vyslaného signálu. K tomuto výpočtu se používají techniky korelace obdržených signálů [9].

2.5.1 Princip

Metody využívající principu TDOA jsou zpravidla používány pokud se lokalizaci snaží provést pouze jedna strana. Zpravidla stačí aby cíl, který se snažíme lokalizovat, vydával nějaký signál a strana provádějící lokalizaci měla dostatečně citlivé senzory, aby tento signál zachytila.

K úspěšné lokalizaci je zapotřebí, aby strana provádějící lokalizaci měla k dispozici pole mikrofonů. Při zachycení senzorů tímto polem má strana provádějící lokalizaci k dispozici pouze zachycené signály. Z těchto signálů lze vypočítat sílu obdrženého signálu a časový rozdíl dopadu signálu na jednotlivé senzory.

Tyto informace jsou ovšem dostačující pro úspěšnou lokalizaci zdroje signálu. Z časového rozdílu dopadu signálu lze spočítat TDOA a na základě tohoto výpočtu lze za použití algoritmů pro lokalizaci na základě TDOA (Hyperbolická lokalizace) vypočítat pozici zdroje signálu.

TDOA oproti TOA nepotřebuje znát přesný čas vyslání signálu, pouze čas příchodu signálu a jeho rychlost. Pokud bude signál přijat alespoň dvěma referenčními body, lze rozdílem časů příchodu tohoto signálu určit poměr vzdáleností zdroje signálu od těchto bodů podle rovnice 2.2.

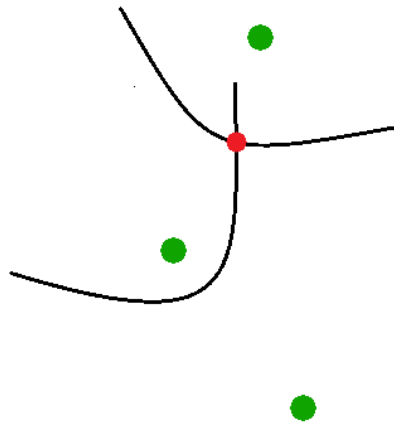
$$\Delta d = c * \Delta t \quad (2.2)$$

kde c je rychlost signálu a Δt je časový rozdíl příchodu signálu. Pokud tuto hodnotu dosadíme do rovnice pro výpočet vzdálenosti mezi zdrojem a cílem (2.3),

$$\Delta d = \sqrt{(x_2 - x)^2 - (y_2 - y)^2} - \sqrt{(x_1 - x)^2 - (y_1 - y)^2} \quad (2.3)$$

kde (x_1, y_1) a (x_2, y_2) jsou známé pozice senzorů, které přijaly signál, dostaneme ve 2D prostoru úpravou této rovnice rovnici hyperboly, kde senzory tvoří ohniska hyperbol [8].

Pokud máme k dispozici více těchto hyperbol, lze spočítat pozici cíle vůči těmto senzorům. Na obrázku 2.2 lze vidět příklad výpočtu pozice pomocí TDOA.



Obrázek 2.2: Schéma výpočtu pozice s využitím TDOA. Zeleně jsou označeny přijímače, červeně je označena pozice zdroje signálu a černě jsou vyznačeny hyperboly, na kterých leží zdroj vypočítané podle poměru času příchodu.

2.6 Time of Arrival - TOA

Time of arrival (dále jen TOA) je jednoduchá technika používaná v mnoha dnešních aplikacích (jedna z nejvýznamnějších je GPS). Tato metoda se používá pokud spolu komunikují dvě strany, jejichž cílem je určení pozice. Pro určení této pozice spolu musí tyto strany spolupracovat, aby došlo k přesnému určení polohy. Určování polohy probíhá pomocí orientačních bodů, které vysílají signál opatřený časovými razítky. Pro přesné určení polohy je potřeba mít několik těchto orientačních bodů.

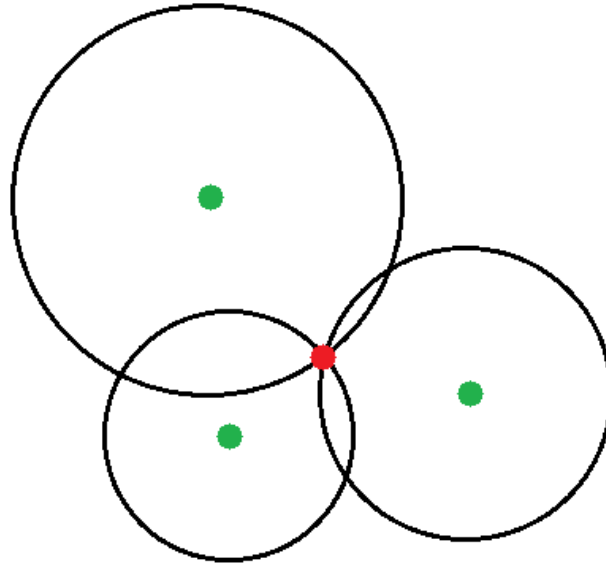
2.6.1 Princip

Tato metoda je založená na měření doby mezi vysláním signálu a jeho přijetím. Pokud známe čas, kdy byl signál odeslán a čas, kdy byl signál přijat, lze ze znalosti rychlosti šíření signálu a doby letu vypočítat vzdálenost vysílače a přijímače [5] [3]. Tuto vzdálenost, d , je možné vypočítat ze vztahu 2.4.

$$d = c * (t_{cas_prijmu} - t_{cas_odeslani}) \quad (2.4)$$

kde c je rychlost signálu. V 2D prostoru tato rovnice udává vzdálenost, která je reprezentována kružnicí, ve 3D je to koule. Pokud jsme schopni tuto vzdálenost vypočítat vůči více zdrojům, jejichž poloha je předem známá, můžeme jednoduše vypočítat pozici vůči těmto zdrojům.

Příklad rozestavení zdrojů signálu a cíle, který počítá svoji polohu, lze vidět na obrázku 2.3.



Obrázek 2.3: Schéma výpočtu pozice s využitím TOA. Zelené body představují přijímače, červený bod představuje zdroj signálu a kružnice představují vzdálenost, kterou signál urazil.

Při lokalizaci s použitím TOA není potřeba pole senzorů. Pro úspěšnou lokalizaci stačí pouze jeden senzor, je ovšem potřeba znát přesnou polohu vysílačů a přesná časová razítka. Na základě těchto časových razítek lze spočítat dobu letu signálu a na základě této hodnoty je možné určit přesnou polohu

Na rozdíl od TDOA, TOA tedy vyžaduje spolupráci více účastníků. Při použití metody TOA se určuje poloha přijímací strany, nikoliv vysílací. Poloha přijímače se určuje na základě signálů z vysílačů.

2.7 Použití TOA a TDOA

Přesto že TOA a TDOA mají společný cíl, lokalizaci, jedná se o zcela odlišné metody. Obě metody potřebují pro lokalizaci senzory a zdroje signálu. Použití těchto prvků se ovšem výrazně liší.

2.7.1 Použití TOA

TOA používá množství zdrojů signálu a účastník, který se snaží vypočítat svoji polohu používá pouze jeden přijímací senzor. Při použití této metody je tedy použito více vysílačů a pouze jeden senzor.

Asi nejznámější použití této lokalizační metody je v **Global Position System** (dále jen GPS). Zde jsou vysílače umístěny na oběžné dráze okolo planety Země a cíle počítající svoji polohu jsou na povrchu.

2.7.2 Použití TDOA

TDOA se používá pro výpočet polohy vysílače signálu. U této metody provádí lokalizaci pouze jedna strana. Lokalizace s použitím TDOA používá pole senzorů a pouze jeden vysílač signálu.

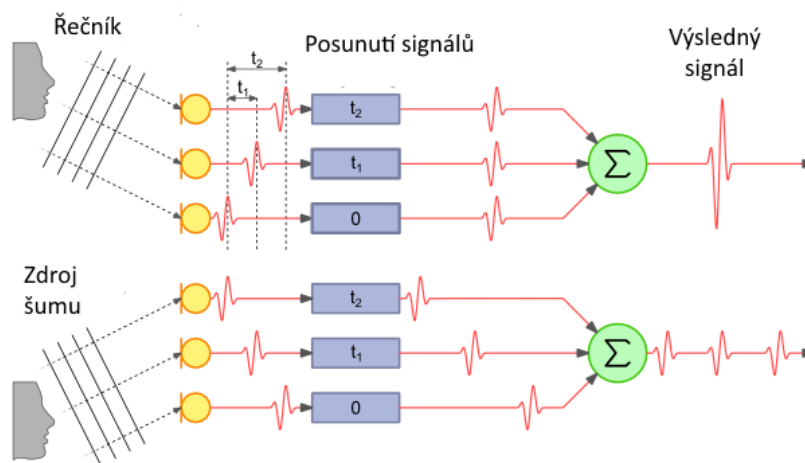
Tato metoda se používá např. u pasivního hyperbolického lokátoru **Tamara** vyvinutého armádou ČR. Tamara se skládá z počtu senzorů, které na základě zachyceného signálu počítají polohu zdroje signálu.

2.8 Beamforming

Beamforming je technika používaná společně s mikrofonními poli za účelem pořízení nahrávky s větší kvalitou signálu. Spočívá ve vytváření tzv. "paprsku" (beam), který se postupně otáčí za pozicí zdroje zvuku. Nahrávka zdroje zvuku v této oblasti paprsku se potom vytvoří složením nahrávek ze všech mikrofónů pomocí současné orientace paprsku.

Směrování paprsku probíhá staticky, protože mikrofonní pole jsou většinou bez jakýchkoliv pohyblivých částí. Nasměrování paprsku je vektor, který udává posun signálu z jednotlivých mikrofónů v čase. S použitím tohoto vektoru se pak posunuté signály sečtou. Po sečtení těchto signálů se správným posunutím se zvýší intenzita požadovaného signálu a intenzita veškerého šumu se sníží nebo se vyruší úplně.

Techniku beamforming lze vidět na obrázku 2.4, kde jsou patrné jednotlivé signály, které po sečtení se správným posunutím vytvoří výsledný zesílený signál.



Obrázek 2.4: Na obrázku lze vidět jednotlivé signály, které sečtením tvoří výsledný signál který obsahuje méně šumu (nahore) a šum, který se po sečtení zredukuje (dole).⁴

Beamforming se používá k prostorovému filtrování signálu. To spočívá v tom, že zaměříme paprsek na nějakou část prostoru a na tuto část se "soustředíme". Beamforming je jedna z nejjednodušších metod prostorového filtrování. Prostorové filtrování je výhodné zejména v situacích, kdy je v prostoru, ve kterém provádíme lokalizaci, velké množství zdrojů šumu, které ovšem nejsou v těsné blízkosti zdroje, jehož pozici se snažíme vypočítat [12].

2.9 Křížová korelace

Křížová korelace slouží k výpočtu koeficientu míry podobnosti. V tomto případě se pomocí korelace bude počítat míra podobnosti dvou signálů. Pokud je v prostředí přítomen zdroj nějakého signálu a my tento signál zachytili polem mikrofonů, mezi nahrávkami, které pořídily jednotlivé mikrofony, existuje určitá korelace. Tuto korelaci lze využít k výpočtu časového posunu mezi signály (TDOA).

Nguyen a Pham (2011) [7] popisují korelaci rovnicí 2.5 a dále ji dělí podle použitých metod pro výpočet vah.

$$R_{x_1x_2}^{GCC}(\tau) = \int_{-\infty}^{\infty} \Psi(\omega) X_1(\omega) X_2^*(\omega) e^{j\omega\tau} d\omega \quad (2.5)$$

Časový posun je zde spočítán pomocí výpočtu inverzní Fourierovy transformace spektra obdržených signálů $X_1(\omega) X_2^*(\omega)$ ovlivněných váhovou funkcí $\Psi(\omega)$.

Dále se výpočty korelace liší na základě výpočtu váhové funkce.

2.9.1 General cross-correlation(GCC)

Pokud nastavíme váhovou funkci $\Psi(\omega)$ na 1, jedná se o klasickou korelaci [7] (Classical Cross-Correlation) popsanou rovnicí 2.6.

⁴Obrázek převzat z <https://arunraghavan.net/2016/06/beamforming-in-pulseaudio/>

$$R_{x_1x_2}^{CC}(\tau) = \int_{-\infty}^{\infty} X_1(\omega)X_2^*(\omega)e^{j\omega\tau} d\omega \quad (2.6)$$

Tímto postupem pouze sledujeme při jakém časovém posunu jsou si signály nejpodobnější. Při takovém postupu ovšem ignorujeme případný šum, který se v signálech vyskytuje. Při použití této metody je tedy potřeba se pokusit odstranit šum ještě před provedením filtrování.

2.9.2 Smoothed coherence transform (SCOT)

Smoothed Coherence Transform je definována způsobem výpočtu vah popsáním v rovnici 2.7.

$$\Psi(\omega) = \frac{1}{|X_1(\omega)||X_2(\omega)|} \quad (2.7)$$

Při použití těchto vah dostaneme výslednou rovnici popsanou vztahem 2.8. Tato metoda je odolnější proti případnému šumu a lépe se vypořádá se změnami intenzity přijatého signálu [7].

$$R_{x_1x_2}^{SCOT}(\tau) = \int_{-\infty}^{\infty} \frac{X_1(\omega)X_2^*(\omega)}{|X_1(\omega)||X_2(\omega)|} e^{j\omega\tau} d\omega \quad (2.8)$$

2.9.3 Phase transform (PHAT)

Metoda Phase Transform počítá váhovou funkci podle vztahu popsaného v rovnici 2.9.

$$\Psi(\omega) = \frac{1}{|X_1(\omega)X_2^*(\omega)|} \quad (2.9)$$

Dosazením vah do rovnice pro korelaci dostaneme vztah 2.10.

$$R_{x_1x_2}^{GCC}(\tau) = \int_{-\infty}^{\infty} \frac{X_1(\omega)X_2^*(\omega)}{|X_1(\omega)X_2^*(\omega)|} e^{j\omega\tau} d\omega \quad (2.10)$$

Tato metoda dává lepší výsledky než obě předchozí metody pro výpočet TDOA s ohledem na množství šumu v signálu.[7].

2.9.4 Výpočet časového posunu signálů

Křížová korelace je velmi užitečná při výpočtu časového posunu mezi dvěma nebo více signály. Pokud známe vzorkovací frekvenci porovnávaných signálů, lze relativně přesně spočítat časový posun těchto signálů. Pro zjištění tohoto časového posunu je potřeba prvně spočítat korelaci pro všechna možná posunutí těchto dvou signálů vůči sobě pomocí výše zmíněných metod. Po výpočtu stačí pouze najít index posunu, kde byla míra podobnosti nejvyšší, tedy kde byla hodnota korelačního indexu nejmenší.

Při znalosti indexu posunu signálů a vzorkovací frekvence lze vypočítat časový posun podle vztahu popsaného rovnicí 2.11.

$$\frac{1}{F} * \Delta d \quad (2.11)$$

kde F je známá vzorkovací frekvence a Δd je index posunu signálů.

Tímto způsobem lze vypočítat hodnotu TDOA a pomocí této hodnoty provést samotnou lokalizaci.

2.9.5 Hraniční hodnoty při výpočtu indexu posunu

U posuvu signálů je ovšem žádoucí abychom signály neposouvali do takové míry, že se signály budou překrývat jen v malém procentu hodnot. Existuje zde možnost, že v okrajových hodnotách, kde se náhodou vyskytnou např. nulové hodnoty bude vypočítaná míra korelace nejvyšší, i když reálný index posuvu se bude vyskytovat na jiném místě. Z tohoto důvodu je potřeba definovat minimální počet vzorků, které se budou v korelaci porovnávat. Volba tohoto prahu je ovšem problematická. Chceme totiž při korelaci porovnávat co největší rozsah indexů, zároveň je ovšem zapotřebí zamezit **false positive** výsledkům. Tato prahová hodnota se může měnit s různými délkami zpracovávaných signálů a neměla by být menší než 10% z celkové délky signálů.

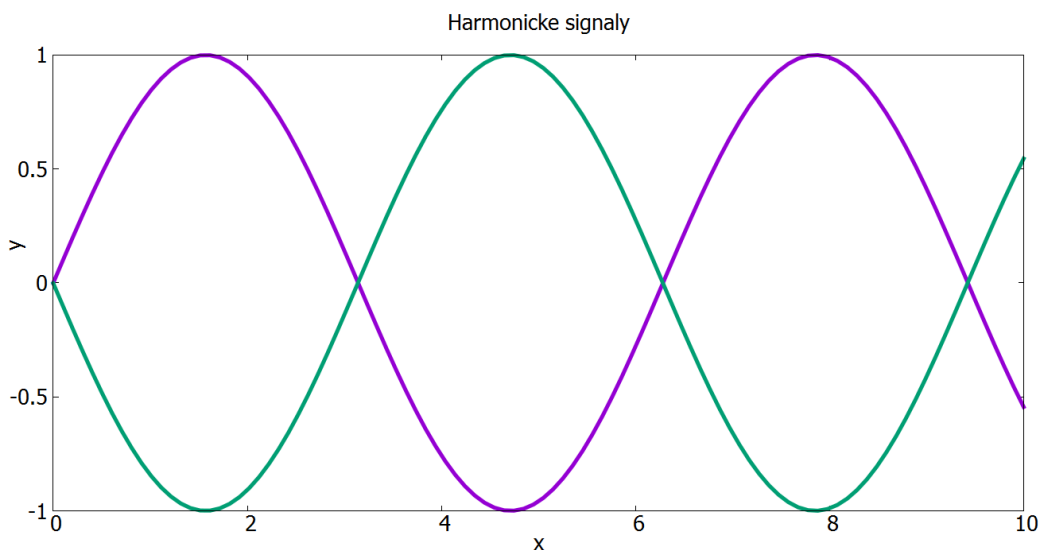
V praxi se volbou tohoto prahu určuje prostor pokrývaný mikrofonním polem. Pokud bude tato prahová hodnota vysoká, budeme provádět korelaci pouze v úzkém prostoru před mikrofonním polem. Naopak při vyšších hodnotách tohoto prahu se zvýší i rozsah sensorového pole.

2.9.6 Nedostatky korelace

Určování posunu dvou signálů v čase pomocí korelace má několik nevýhod. Nevýhoda korelace spočívá v tom, že klasická metoda korelace nebere ohled na případný šum obsažený v signálech. Před samotou korelací je potřeba případný šum odfiltrovat, nebo přijmout případné nepřesnosti.

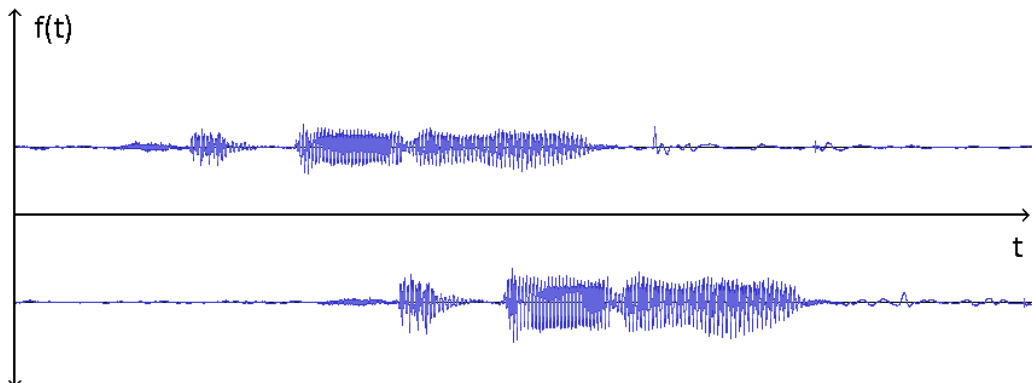
Dalším problémem je korelace harmonického signálu. Korelace využívá rozdílů v signálech, které se nepravidelně mění v čase. Pokud je signál periodický a má stejnou periodu (zde předpokládáme že ano, protože se jedná o tentýž signál, který je pořízen více mikrofony), tyto rozdílly se zde nevyskytují a není podle čeho korelovat.

Na obrázku 2.5 lze vidět příklad harmonických signálů. Zde není zjevné, jestli se tyto signály vůči sobě zpozdily o jednu nebo více period.



Obrázek 2.5: Na obrázku lze vidět dva časově posunuté periodické signály. Nelze s určitostí říct jestli má být zelený signál posunut doleva nebo doprava.

Na obrázku 2.6 lze vidět nahrávku lidské řeči. Zde je jasně vidět, že signály jsou téměř stejné, pouze jsou vůči sobě zpožděné.



Obrázek 2.6: Na obrázku lze vidět dvě nahrávky lidské řeči. V tomto případě bude korelace těchto signálů relativně jednoduchá, i pouhým okem lze vidět, že jsou si tyto signály velice podobné.

2.10 Hyperbolická metoda určení pozice

Hyperbolická metoda je založená na skutečnosti, že pokud vypočítám časový rozdíl dopadu signálu na senzory, je absolutní hodnota tohoto rozdílu konstantní po celé délce hyperboly. Rovnice hyperboly, jejíž hlavní osa je rovnoběžná s osou x , je možno vidět na rovnici 4.5:

$$\frac{(x - m)^2}{a^2} - \frac{(y - n)^2}{b^2} = 1 \quad (2.12)$$

kde $[x, y]$ je libovolný bod ležící na hyperbole, a je délka hlavní poloosy, b je délka vedlejší poloosy a m a n jsou posunutí středu hyperboly. Pokud je střed hyperboly stejný jako střed souřadného systému, pak $m = n = 0$. Při určování pozice si mohou dovolit zvolit střed souřadného systému libovolně, mohou tedy dosadit za n nulu.

K určení rovnice hyperboly je zapotřebí znát délky hlavní a vedlejší poloosy. Tyto hodnoty lze spočítat ze znalosti polohy ohnisek a vrcholů hyperboly. Souřadnice ohnisek jsou známá, jsou to souřadnice senzorů. Pozici vrcholů lze spočítat pokud vypočítáme vzdálenost, Δd , kterou zvuk urazil za časový rozdíl dopadu signálů na senzory. Tuto vzdálenost lze vypočítat z rovnice 2.13:

$$\Delta d = \Delta t * v_{zvuk} \quad (2.13)$$

kde Δt je časový rozdíl dopadu a v_{zvuk} je rychlost zvuku. Z tohoto důvodu je nutné znát přesnou rychlost zvuku v daném prostředí.

Pokud známe Δd a souřadnice ohnisek, lze vypočítat i souřadnice vrcholů hyperboly. Víme, že tyto body leží na přímce mezi ohnisky, tedy y souřadnice je nulová. Také víme, že platí vztah 2.14 pro výpočet délky hlavní poloosy.

$$\Delta d = 2a \quad (2.14)$$

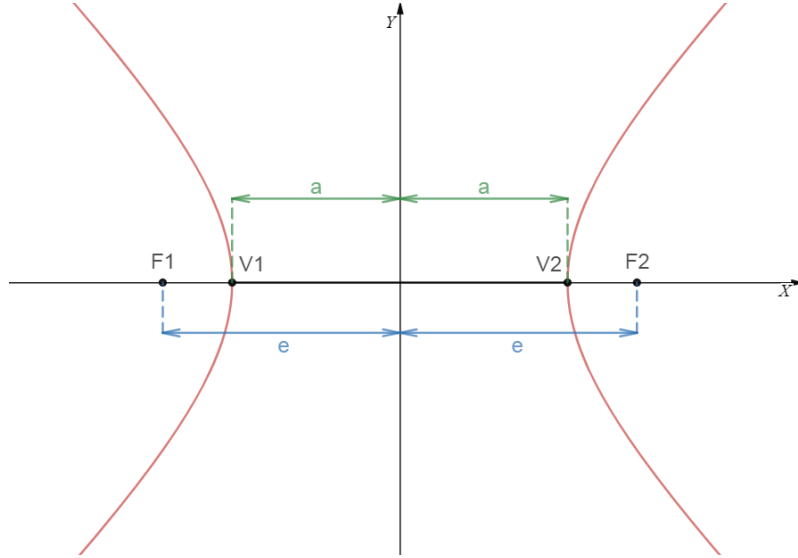
Souřadnice vrcholů hyperboly jsou $[a, 0]$ a $[-a, 0]$. Tímto dostaneme dva body ležící mezi ohnisky a to jsou vrcholy hyperbol. Excentricita hyperboly, e , je vzdálenost mezi středem hyperboly a ohniskem a bude využita k výpočtu délky vedlejší poloosy.

Pro výpočet délky vedlejší poloosy lze využít následující rovnice:

$$a^2 + b^2 = e^2 \quad (2.15)$$

kde a a b jsou délky hlavní a vedlejší poloosy a e je excentricita, jinak nazývána vzdálenost mezi ohnisky.

Pro přesnější představu je vše znázorněno na obrázku 2.7.



Obrázek 2.7: Na obrázku lze vidět jak souvisí délka hlavní poloosy (a) a excentricita (e) se souřadnicemi ohnisek ($F1, F2$) a vrcholů hyperboly ($V1, V2$).

Těmito výpočty jsou známy všechny proměnné pro vytvoření rovnice hyperboly, na které leží zdroj signálu. Pro N senzorů lze vytvořit $\binom{N}{2}$ hyperbol, pokud budeme navzájem porovnávat všechny mikrofony.

2.10.1 Průsečík hyperbol

Předpokládejme, že máme k dispozici rovnice dvou hyperbol:

$$\frac{(x_1 - n_1)^2}{a_1^2} - \frac{y_1^2}{b_1^2} = 1 \quad \frac{(x_2 - n_2)^2}{a_2^2} - \frac{y_2^2}{b_2^2} = 1$$

Lze vidět, že posunutí v ose y jsou nulová. Tuto úpravu si můžeme dovolit protože si můžeme zvolit střed souřadného systému v bodě $[n, 0]$. Zde je střed zvolen mezi dvěma mikrofony, jejichž souřadnice budou zároveň tvořit i ohniska. Průsečík hyperbol lze spočítat tak, že z obou rovnic vyjádříme některou ze souřadnic a vyřešíme vzniklou kvadratickou rovnici 2.16

$$b_1^2 \frac{(x - n_1)^2}{a_1^2} - b_1^2 = b_2^2 \frac{(x - n_2)^2}{a_2^2} - b_2^2 \quad (2.16)$$

kořeny z této rovnice lze dosadit do rovnic hyperbol a dostaneme příslušné body průniku.

2.11 Optimalizovaná hyperbolická metoda pro určení polohy

Zatímco klasická metoda hyperbolické lokalizace využívá analytického řešení rovnic, které je v počítači značně pomalé, tato metoda se zaměřuje na rychlost za cenu menší přesnosti.

2.11.1 Princip

Metoda je založená na tom, že pokud máme 2 pevně daná ohniska, lze pouze změnou délky hlavní poloosy mezi nimi vytvořit nekonečné množství hyperbol. To je možné pomocí následujícího vztahu:

$$a^2 + b^2 = e^2$$

po vyjádření b a dosazení do rovnice hyperboly dostaneme rovnici 2.17

$$\frac{x^2}{a^2} - \frac{y^2}{e^2 - a^2} = 1 \quad (2.17)$$

tvoření rovnic spočívá v posouvání hodnoty a (hlavní poloosa), protože e zůstává po celou dobu konstantní. Tyto hyperboly rozdělí prostor mezi ohnisky na několik intervalů na ose x . Tyto hraniční hodnoty lze předpočítat po konstantních krocích na ose y , protože pozice mikrofonů (ohnisek) je konstantní. Místo rovnice hyperboly dostaneme interval ohraničený dvěma hyperbolami, jejichž hraniční hodnoty již známe.

Po vypočítání rozdílu z dat z mikrofonů dostaneme délku hlavní poloosy hyperboly, na které leží zdroj zvuku. Tato hyperbola zjevně leží v některém z intervalů mezi ohnisky. Pokud vypočítáme poměrnou vzdálenost od hraničních hodnot intervalu ve kterém leží tato hyperbola, tento poměr se nebude měnit se zvyšující se hodnotou y . Pokud máme dostatečné množství mikrofonů na vytvoření více hyperbol, lze pomocí těchto intervalů vypočítat průsečík těchto hyperbol. Na tento průsečík se lze ovšem dívat jako na průsečík dvou přímek, pokud využijeme faktu, že se poměrná vzdálenost mezi intervaly nemění, lze tyto body na intervalu spojit se sousedním krokem na ose y a dostat tak hledanou přímku. Pokud nalezneme průsečík těchto přímek, nalezneme i hledanou polohu zdroje zvuku.

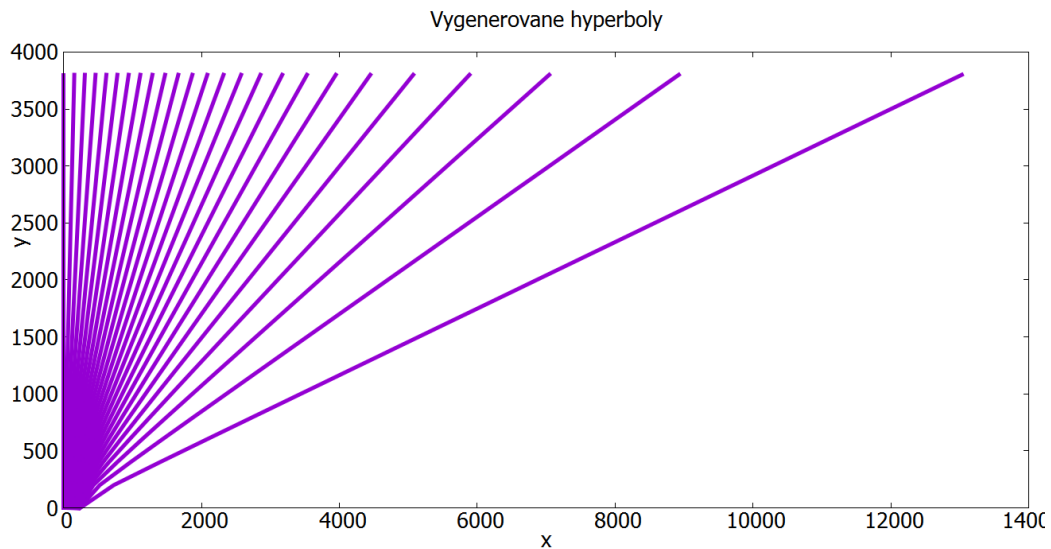
Předpočítané hodnoty hyperbol lze vidět na obrázku 2.8

2.11.2 Výhody

Tato metoda je rychlejší než analytické řešení průsečíku dvou hyperbol pomocí soustavy rovnic. Pro výpočet průsečíku není potřeba žádné umocňování, které je značně pomalé. Další výhodou je rychlost díky tomu, že hraniční hodnoty jsou již předpočítané, což dále redukuje počet matematických operací, které je potřeba vykonat k výpočtu průsečíku.

2.11.3 Nevýhody

Nevýhodou této metody je menší přesnost, než u analytického výpočtu. Již v ideálních podmínkách má tato metoda odchylku od reálného průsečíku, které závisí na vzdálenosti od mikrofonního pole. Tato nepřesnost je způsobena zejména nepřesnostmi vyplývajícími z použití poměrů namísto rovnic hyperbol. Další nepřesnost je způsobena při výpočtu odlehých hodnot. Experimenty bylo zjištěno, že pokud je granularita intervalů nízká a hraničních hyperbol je menší počet, pro zdroje zvuku, které jsou blízko pole, ale v odlehle části prostoru před mikrofonním polem, je tato metoda značně nepřesná. Tuto skutečnost lze ovšem



Obrázek 2.8: Na obrázku lze vidět předpočítané hyperboly pro rozestup mikrofónů 500 mm vynesené do grafu.

zanedbat, protože ji lze řešit zvýšením počtu intervalů, nebo vytyčením intervalu přesnosti mikrofonního pole.

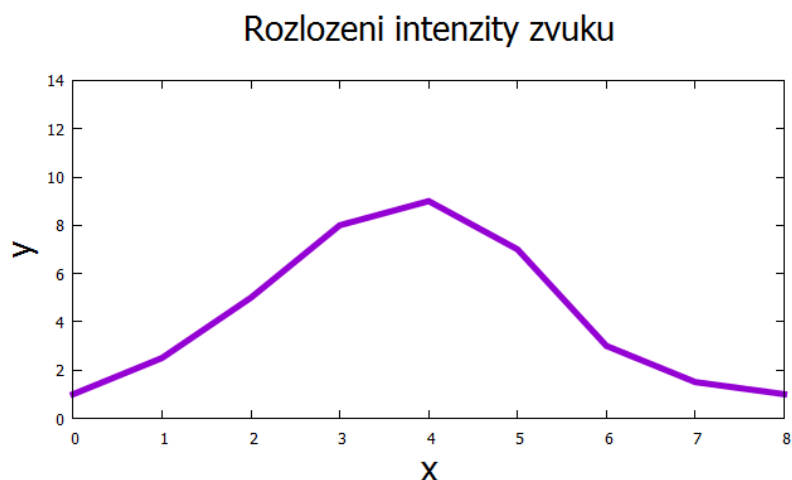
2.12 Metoda využívající síly signálu

Tato metoda využívá faktu, že intenzita signálu s uraženou vzdáleností klesá. Metoda využívá tohoto faktu k odhadu pozice zdroje zvuku. Pro výpočet pozice pomocí této metody není potřeba znát přesný signál a nemusí se tedy řešit problémy spojené s filtrací. Důležité je ovšem znát intenzitu signálů.

2.12.1 Princip

Pokud se v prostoru před mikrofonním polem nachází zdroj zvuku a je dostatečně silný aby jej mikrofony zachytily, lze z nahraného signálu vypočítat intenzitu. Víme ovšem, že zdroj musí mít rozdílnou vzdálenost od jednotlivých mikrofónů. Díky tomuto faktu budou mít signály nahrané jednotlivými mikrofony rozdílnou intenzitu.

Pokud tuto intenzitu vyneseme do grafu 2.9, kde na x ose jsou jednotlivé mikrofony tak, jak jsou rozestavené na rameni mikrofonního pole, lze vidět, že se v grafu objeví maximum.



Obrázek 2.9: Na obrázku lze vidět rozložení intenzity zvuku při dopadu na jednotlivé mikrofony.

Toto maximum znamená, že se zdroj zvuku nachází před daným mikrofonem, protože se nachází nejbližší tomuto mikrofonu. Lze vidět, že oběma směry intenzita klesá se zvyšující se vzdáleností zdroje zvuku od mikrofonů.

2.12.2 Výhody

Tato metoda je velmi jednoduchá na zpracování. Nepotřebuje filtrovat signál ani jej jiným způsobem upravovat. Další výhodou je její rychlost. Po spočítání intenzity je jediný problém nalezení maxima, který jsme schopni řešit v čase $O(n)$.

2.12.3 Nevýhody

Tato metoda má ovšem velmi výrazné nevýhody. Jedna z nevýhod je, že metoda pracuje pouze v jedné ose. Na rozdíl od předchozích hyperbolických metod, kde jsme z řady mikrofonů mohli vypočítat jak x , tak y souřadnici, tato metoda je schopna vypočítat pouze jednu souřadnici. Metoda není schopna spočítat vzdálenost zdroje pouze pomocí jednoho ramene s mikrofony. Další nevýhodou je velká závislost na orientaci zdroje zvuku. Tato metoda dává posunuté výsledky pokud má např. řečník otočenou hlavu doleva či doprava.

I přesto, že má tato metoda značné nevýhody, lze ji uplatnit např. jako kontrolu při výpočtu pomocí ostatních metod.

Kapitola 3

Pokusné mikrofonního pole

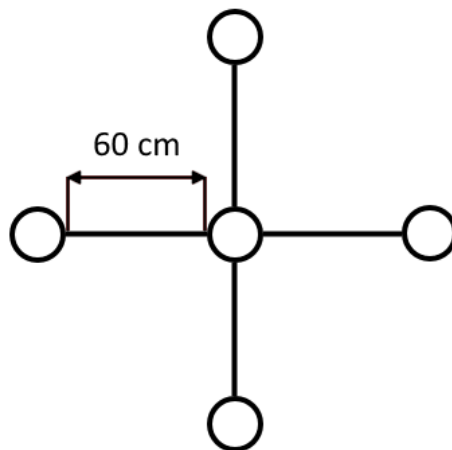
V této kapitole jsou popsány detaily konstrukce prototypu mikrofonního pole, jeho parametry a důvody ke zvolení těchto parametrů. Dále zde bude popsána implementace použitých algoritmů a jejich úpravy.

Toto mikrofonní pole bylo sestaveno jako nástroj pro pokusy s algoritmy pro lokalizaci zdroje zvuku, pokusy se vzorkovací frekvencí a pro optimalizaci komunikace mezi mikrofonním polem a počítačem.

Při pokusy s mikrofonním polem jsem experimentálně zjišťoval optimální parametry pro co nejlepší funkčnost pole. Důraz jsem kladl zejména na přesnost určování pozice zdroje zvuku, rychlost posílání dat z desky do PC a zpracování dat z mikrofonního pole.

3.0.1 Rozestavění mikrofonů

Mikrofonní pole bylo zkonstruováno jako rovinné pole. Skládá z pěti mikrofonů rozestavěných do kříže jak je znázorněno na obrázku 3.4.



Obrázek 3.1: Rozestavění mikrofonů v poli. Jedná se o pole umístěné v jedné rovině. Mikrofonní pole je zde zobrazeno při pohledu zepředu. Na obrázku je vyznačena i vzdálenost mezi jednotlivými mikrofony. Tato vzdálenost je stejná pro všechna ramena kříže.

Vzdálenost mezi mikrofony jsem zvolil 60 cm. Tuto vzdálenost jsem zvolil jako kompromis mezi poměrem vzdálenosti mezi mikrofony k vzorkovací frekvenci a celkovou velikostí mikrofonního pole, protože mikrofony, které tvořily toto pole, měly omezený dosah. Pokud

by tato vzdálenost byla příliš velká mohlo by doházet k tomu, že pokud by řečník stál u některého z okrajových mikrofonů nemusel by jej mikrofon umístěný na protějším rameni pole slyšet.

3.1 Použitý hardware

Při výběru hardware jsem kladl důraz zejména na dostupnost v ČR. Dalším faktorem byla možnost programování desky v Arduino vývojovém prostředí a tím pádem i součástek kompatibilních s Arduino deskami.

Toto pole bylo sestaveno proto, že jsem v této době neměl k dispozici mikrofonní pole zapůjčované školou.

3.1.1 Mikrofon

Jako mikrofon jsem zvolil **GY-MAX4466 Electret Microphone** (k vidění na obr. 3.2). Jedná se o cenově dostupný analogový mikrofon kompatibilní s deskami Arduino.

Mikrofon je schopný snímat signály o frekvenci 20 Hz - 20 kHz. Jeho napájecí napětí je v rozsahu 2.4V – 5.5V.



Obrázek 3.2: Na obrázku lze vidět použitý mikrofon. Jedná se o analogový mikrofon s nastavitelnou hodnotou gain.²

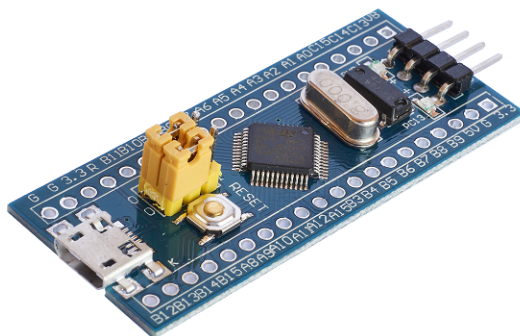
Jediným větším omezením mikrofonů je to, že jejich dosah je značně omezen. Experimentálně jsem odhadl jejich nahrávací vzdálenost při normální síle lidského hlasu na 0.7 m. Pokud je tato vzdálenost větší, je potřeba zvýšit hlasitost nebo nedojde k zachycení vydávaného zvuku. Protože jde ovšem pouze o prototyp, který slouží zejména k ověření funkčnosti algoritmů, je tato vzdálenost dostačující.

3.1.2 Deska s mikrokontrolérem

Jako mikrokontrolér jsem zvolil desku s označením **Blue Pill** vyráběnou společností LeafLabs. Tuto desku jsem vybral zejména kvůli její cenové dostupnosti, výkonu, kompatibilitě s Arduino vývojovým prostředím a A/D převodníkem dostupným na desce.

²Obrázek převzat z <https://laskarduino.cz/vstupni-periferie-cidla/230534-gy-4466-modul-mikrofonu-s-zesilovacem-max4466.html>

Desku lze vidět na obrázku 3.3.



Obrázek 3.3: Na obrázku lze vidět programovatelnou desku Blue Pill.³

Vybrané technické detaily:

- Jádru: Cortex-M3 32-bit
- Operační frekvence: 72 MHz
- Paměti: 64 KB Flash, 20 KB SRAM
- A/D převodník: 12 bit

Desku jsem programoval ve vývojovém prostředí společnosti Arduino. Programoval jsem ji přes ST-link v2 programátor připojený do PC přes USB rozhraní.



Obrázek 3.4: ST link programátor použitý při programování desky.⁴

Jako největší nevýhodu této desky považuji poměrně malou velikost operační paměti RAM, 20 kB. V praxi to znamenalo to, že při vyšších vzorkovacích frekvencích (50 kHz a výše) docházela paměť na uchování celého bloku vzorků. Vzhledem k tomu, že lidská řeč má frekvenční pásmo mezi 80 Hz a 3 kHz ukázalo se jako dostačující vzorkovat data frekvencí 8 kHz. Při použití této frekvence deska stíhala posílat v čase mezi pořizováním jednotlivých vzorků data do PC.

³Obrázek převzat z https://wiki.stm32duino.com/index.php?title=Blue_Pill

⁴Obrázek převzat z <https://www.diyelectronics.co.za/store/other/1183-st-link-v2-programmer-for-stm8-stm32.html>

3.2 Popis rozdělení úloh mezi PC a mikrokontrolér

Protože deska s mikrokontrolérem není dostatečně výkonná na to, aby v ní byl nahraný celý program pro lokalizaci, bylo potřeba rozhodnout, jakou funkci bude vykonávat deska a jakou PC.

Bylo potřeba vzít v úvahu omezené paměťové možnosti mikrokontroléru (20 kB) a taktovací frekvenci jádra (72 MHz). Veškeré výpočetně náročné úkony má proto na starosti PC.

3.2.1 Mikrokontrolér

Primární funkce mikrokontroléru je sběr dat z mikrofonů. Toto je potřeba realizovat přes analogový převodník, protože mikrofony jsou analogové. Na desce mikrokontroléru je k dispozici 12 bitový převodník. Převedené hodnoty z převodníku je tedy nutné reprezentovat nejméně 16-bitově (pokud neuvažujeme uložení v bitových polích). Je ovšem možné tuto hodnotu normalizovat např. na 8 bitů, ovšem tímto procesem se drasticky redukuje kvalita záznamu (z 4096 na 256 hodnot rozlišení).

Po obdržení dat z převodníku je potřeba data odeslat do PC. Odesílání je realizováno přes sériovou linku. Zde je možné použít dva přístupy. Odeslat načtená data okamžitě, nebo data ukládat do paměti a počkat na načtení celého bloku. Volba přístupu záleží hlavně na vzorkovací frekvenci a počtu kanálů. Pokud bude vzorkovací frekvence příliš vysoká nebo pokud bude potřeba obsluhovat větší počet kanálů bude potřeba počkat na načtení celého bloku, protože prodleva mezi jednotlivými vzorky bude moc krátká na zaslání všech načtených dat. Tímto ovšem vznikají další problémy jako např. časové prodlevy mezi navzorkovanými bloky pokud bude v PC implementováno klouzavé okno nebo fakt, že se celý načtený blok nemusí vejít do operační paměti mikrokontroléru.

Při vzorkovací frekvenci 8 kHz a při načítání dat z pěti kanálů byla dostatečná prodleva pro odeslání všech potřebných dat.

3.2.2 PC

Díky většímu výkonu má PC za úkol provádět veškeré výpočetně náročné operace. Po obdržení dat z mikrokontroléru je potřeba provést křížovou korelaci obdržených signálů, pomocí které se vypočítá vzájemné zpoždění. Za pomoci tohoto zpoždění je potom možné pomocí algoritmů spočítat souřadnice zdroje zvuku.

3.3 Křížová korelace

Primárně jsem použil algoritmy, které odhadují pozici pomocí `Time Difference Of Arrival`.

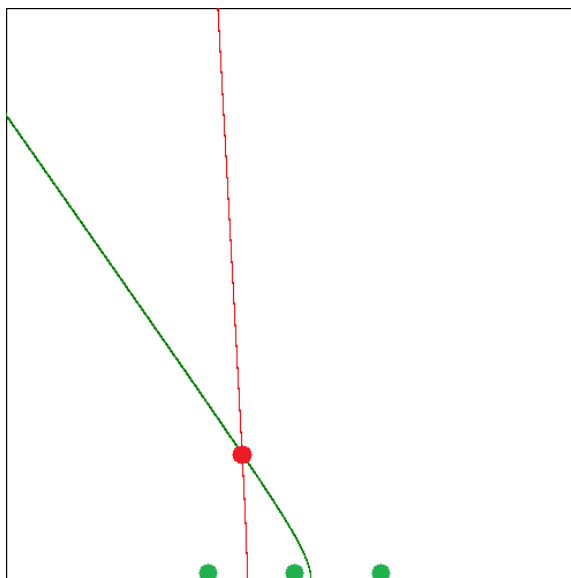
Křížová korelace dvou diskrétních signálů $x[k]$ a $y[k]$ je počítaná podle vzorce 3.1.

$$R_{xy}[k] = \sum_{m=-\infty}^{\infty} |x[m] - y[m - k]| \quad (3.1)$$

kde parametr k je číslo v intervalu $< -\infty, \infty >$ a značí posunutí signálů.

Korelace se počítá s posunem o 1. Není použita žádná heuristika pro skoky o větším kroku, která by mohla potenciálně celý algoritmus značně urychlit. Pro každé takové posunutí se počítá křížová korelace signálu ze dvou mikrofonů a vybere se takové posunutí, kde je hodnota křížové korelace nejmenší, tedy kdy jsou si signály nejpodobnější.

Pro odhad pozice polohy zdroje zvuku používám hyperbolickou metodu. Tento princip lokalizace využívá faktu, že absolutní hodnota rozdílu vzdáleností od dvou ohnisek je stejná na celé hyperbole. Lze využít faktu, že při porovnání signálů ze dvou mikrofonů, kde dostaneme časový rozdíl dopadu těchto signálů, lze při znalosti vzorkovací frekvence jednoduše spočítat tento rozdíl.



Obrázek 3.5: Hyperbolická lokalizace ve 2D prostoru za použití tří mikrofonů umístěných v jedné přímce. Zelené body označují mikrofony, červený bod značí vypočítanou pozici a zelená a červená křivka jsou hyperboly vypočítané z časových rozdílů mezi sousedními mikrofony.

Pokud máme tři mikrofony, dostaneme tímto postupem dvě hyperboly, jak je naznačeno na obrázku [A.1](#). Pro výpočet odhadu pozice je zapotřebí vypočítat průsečík těchto dvou hyperbol.

Kapitola 4

Detaily implementace prototypu

Maple deska, na kterou je napájeno mikrofonního pole, posílá do PC přes sériovou linku navzorkované signály z pěti mikrofónů o vzorkovací frekvenci 8 kHz, které obsahují data o časovém rozmezí 10 ms. V PC je potom vytvořeno klouzavé okno, které uchovává vzorky z posledních 100 ms, kdy nejstarší 10 ms vzorek se přepisuje tím nejnovějším, který byl právě obdržěn z desky. Tento časový interval byl zvolen s ohledem na fyzická omezení lidského pohybu a frekvenci lidského hlasu. Celý algoritmus byl navržen tak, aby byl schopen sledovat pohyb lidí, kdy není potřeba zpracovávat stálý proud dat, protože člověk může dosáhnout jen určité maximální rychlosti. Toto omezení dává prostor pro případné složitější výpočty jako např. filtrace signálu apod.

4.1 Data z mikrofonního pole

Mikrofonní pole nahrává data pomocí pěti mikrofónů. Tato data jsou navzorkována vzorkovací frekvencí 8 kHz. Celkem jsou nahrávány úseky o délce 10 ms.

4.1.1 Objem dat

Při vzorkování se vzorkovací frekvencí 8 kHz a při době nahrávání 10 ms se celkem nahraje 80 vzorků z jednoho mikrofónu. Při nahrávání dat z pěti mikrofónů je tedy k dispozici celkem 400 vzorků. Jeden vzorek je reprezentován dvěma byty, protože A/D převodník, který je na desce k dispozici, převádí data na 12 bitů. Tato hodnota je reprezentována typem `int` o velikosti 16 bitů. Dohromady je tedy v paměti uloženo 800 bytů dat. Tuto velikost je možné zredukovat, pokud by se tato data ukládala do bitového pole. Tato optimalizace by zredukovala paměť potřebnou k uložení dat z 800 na 600 bytů, ovšem i bez této úpravy se dají data pohodlně uložit do paměti a zaslání do PC není tak dlouhá, aby se musela přidávat tato režie.

4.1.2 Přenos dat

V původním návrhu jsem před zasláním počkal až bude k dispozici celý blok dat. Takový přístup se ovšem ukázal jako chybný, protože se ukázalo, že doba potřebná k zaslání takového objemu dat je nezanedbatelná vzhledem k časovým úsekům se kterými pracuji. Ukázalo se že pro navzorkovaných 10 ms je odesílací doba asi 7 ms (tato hodnota byla experimentálně zjištěna). Tato skutečnost představovala problém, protože v klouzavém okně,

kteře bylo uloženo v PC byla mezi jednotlivými 10 ms okny 7 ms prodleva. Tato prodleva ovlivňovala přesnost korelace a bylo nutné použít jinou metodu pro zasílání dat.

Při vzorkovací frekvenci 8 kHz jsou časové rozestupy mezi jednotlivými vzorky 125 μ s. Ukázalo se, že tato doba je dostatečně dlouhá k přečtení hodnot z A/D převodníku a následnému zaslání po sériové lince. Je zde sice možnost, že zasílací doba se může prodloužit a tím se zvětší rozestupy mezi vzorky, ale při experimentech s tímto postupem k tomuto neočekávanému prodlužování nedocházelo. Tato strategie se tedy ukázala jako vhodné řešení tohoto problému.

4.1.3 Komunikační protokol

Při zasílání hodnot se prokládají hodnoty z jednotlivých mikrofonů. Vzorky z jednoho mikrofonu se tedy opakují po 5 hodnotách.

Aby se předešlo jakékoliv chybě při přenosu dat, samotným datům předcházejí uvozující znaky a dalšími uvozovacími znaky je tato posloupnost i ukončena. Tímto způsobem jsem zajistil, aby se předešlo jakýmkoliv konfliktům mezi mikrokontrolérem a PC.

Posloupnosti dat předchází vždy dvojice znaků `##`. Po této uvozující sekvenci následují samotná data, která jsou vždy zakončena symbolem `*`. Teprve po přijetí znaku `*` se s daty dále pracuje. Pokud nedojde k přijetí `*` data jsou ignorována a to i za předpokladu, že počet samotných datových bytů odpovídá očekávanému počtu.

Celá komunikační sekvence vypadá následovně:

```
##mic1_byte[0] mic1_byte[1] mic2_byte[0] ... mic5_byte[158] mic5_byte[159]*
```

4.1.4 Filtrace signálu

Lidská řeč má relativně malý rozsah frekvenčního spektra. Pokud se tedy v signálu lidské řeči vyskytnou abnormálně vysoké nebo nízké frekvence, je velmi pravděpodobné, že jde o šum. Z tohoto důvodu před provedením korelace tyto nežádoucí frekvence odfiltruji pomocí `band-pass` filtru. K filtrování jsem použil `Chebyshev FIR` filtr ¹.

Tento filtr má za úkol odstranit vysoké frekvence, které se v nahrávce nemohou vyskytnout v lidské řeči, a stejně tak i velice nízké frekvence.

Celý proces filtrování je prováděn softwarově na PC. Samotné mikrofony neobsahují žádné hardwarové filtry. Stejně tak není filtrování prováděno na desce, která na tyto operace nemá potřebný výkon. Dále uvedené operace jsou všechny prováděny přímo v PC.

4.1.5 Normalizace střední hodnoty signálu

Pokud odhaduji pozici jednoho zdroje zvuku, mikrofony zachytí signál o různé síle. Toto je způsobeno rozdílnými vzdálenostmi mikrofonů od zdroje zvuku. Před samotnou korelací jsem znormalizoval střední hodnoty signálů. Prvně jsem spočítal střední hodnotu d signálu f .

$$d = \frac{1}{N} \sum_{k=1}^N f[k]$$

Poté jsem tuto hodnotu použil k tomu, abych upravil střední hodnoty signálů na 400. Tato konstanta nemá žádný speciální význam, chtěl jsem se akorát vyhnout počítání s

¹Knihovna dostupná ke stažení na <https://cardinalpeak.com/blog/a-c-class-to-implement-low-pass-high-pass-and-band-pass-filters/>

plovoucí desetinou čárkou. Tímto jsem celou operaci urychlil a rozhodl se přijmout případné nepřesnosti ve výpočtech.

4.2 Detekce zvuku

Všechny mikrofony v tomto poli nahrávají zvuk neustále. Tedy i v případě, že je v místnosti naprosté ticho. Pokud se ovšem budeme snažit o korelaci dvou signálů, ve kterých je přítomen jenom šum, dostaneme naprosto náhodnou polohu zdroje zvuku v prostoru. Pro předejití tomuto jevu je implementován práh, který pokud je překročen, zahájí se korelace, jinak jsou signály ignorovány a čeká se na přijetí dalšího 10 ms segmentu.

Tato detekce je implementována jako nalezení maximální energie v obdrženém signálu. Pokud je u alespoň jednoho signálu nalezeno maximum, které překročí předem stanovenou hodnotu prahu, má smysl pokoušet se o korelaci mezi těmito signály. Pokud ne, předpokládám, že se v nahrávce vyskytuje pouze "ticho"s případným šumem z okolí.

4.2.1 Zpracování signálu

Pro výpočet pozice jsem použil pomocný výpočet TDOA. Pro výpočet TDOA jsem použil křížovou korelaci. Signál ze středového mikrofону jsem použil jako referenční a pomocí korelace s ostatními signály jsem vypočítal časový posun vůči tomuto mikrofónu.

4.3 Křížová korelace

Při porovnávání dvou signálů o délce 800 vzorků se spolu porovnává vždy minimálně 200 vzorků z každého signálu.

Tímto postupem jsem zamezil **false-positive** výsledkům, kdy by se mezi sebou porovnával menší počet hodnot, jejichž průměrná odchylka by byla téměř nulová a tímto by se tento výsledek považoval za nejlepší i přesto, že hledaný výsledek by mohl mít větší odchylku. Při použití této strategie se musí provést celkem 1200 výpočtů průměrných odchylek pro signály ze 2 mikrofónů. Takové sekvence výpočtů se provádí celkem čtyři. Provádí se korelace každého okrajového mikrofónu se středovým mikrofónem.

4.4 Výpočet polohy

K výpočtu pozice zdroje zvuku používám hyperbolickou metodu. Pomocí této metody z výsledku korelace mezi dvěma signály vytvořím hyperbolu, na které leží odhadovaný zdroj zvuku. Pokud mám k dispozici těchto hyperbol více, je pravděpodobné, že na jejich průsečíku popř. v jeho okolí leží zdroj zvuku. Čím více hyperbol je k dispozici, tím přesnější tato metoda bude. Pokud se více hyperbol neprotne v jednom bodu, je možné např. vypočítat těžiště všech bodů popř. přiřadit těmto bodům váhy podle výsledků korelace.

4.5 Časově hyperbolická metoda lokalizace

Při výpočtu polohy jsem se soustředil na nalezení pozice ve 2D prostoru. Protože jsou ramena mikrofónního pole na sebe kolmá, lze úlohu lokalizace ve 3D prostoru rozdělit na dva problémy lokalizace ve dvou 2D rovinách jež jsou na sebe kolmé.

V každé rovině se nachází tři mikrofony, pro každou rovinu koreluji pouze dva okrajové mikrofony se středovým, provádím tedy průsečík dvou hyperbol pro jednu plochu.

Průsečík počítám po zjištění rovnic dvou hyperbol, kde postupně vyjádřím proměnné y z rovnic

$$\frac{(x_1 - n_1)^2}{a_1^2} - \frac{y_1^2}{b_1^2} = 1 \quad \frac{(x_2 - n_2)^2}{a_2^2} - \frac{y_2^2}{b_2^2} = 1$$

a vypočítám jednu společnou souřadnici podle rovnice 4.1.

$$b_1^2 \frac{(x_1 - n_1)^2}{a_1^2} - b_1^2 = b_2^2 \frac{(x_2 - n_2)^2}{a_2^2} - b_2^2 \quad (4.1)$$

Vypočítanou souřadnici pak lze dosadit do jedné ze dvou původních rovnic a tímto způsobem lze najít průsečík obou hyperbol.

Pokud dostaneme více bodů, lze vyloučit ten, jehož souřadnice nedává smysl. Jeden z předpokladů je, že cíl lokalizace musí být před mikrofonním polem a tedy pokud vyjde y souřadnice záporná můžeme ji vyloučit.

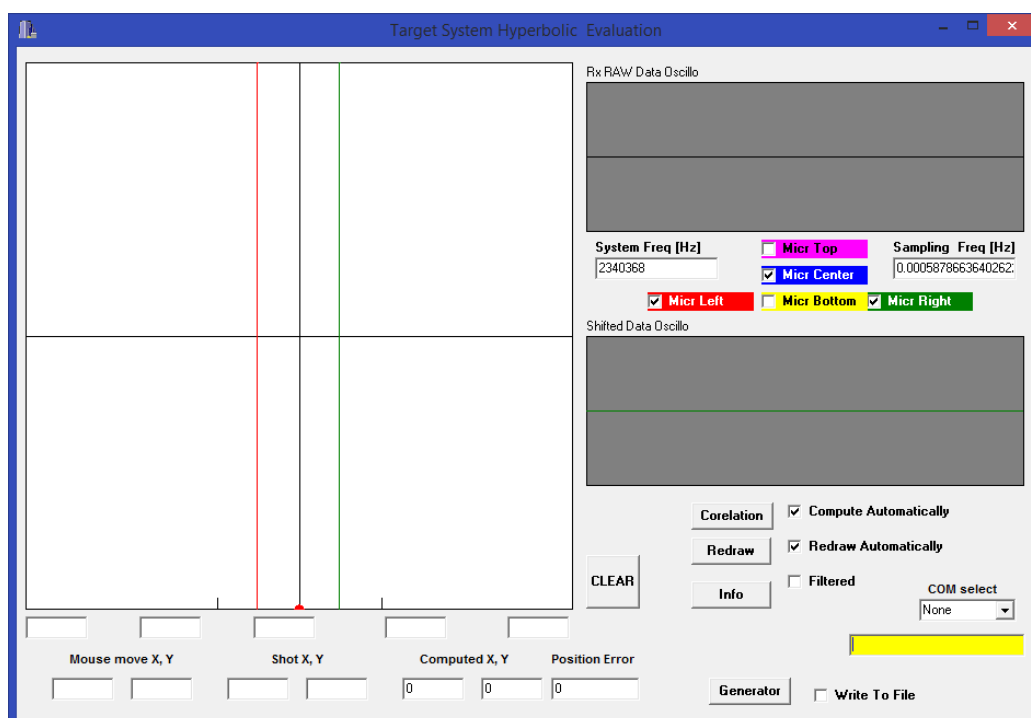
4.5.1 Filtrování pozice

Při použití pouze tří mikrofónů k odhadu lokalizace zdroje zvuku často docházelo k chybnému určení pozice. Tato chyba může vzniknout řadou různých vlivů, zejména nekvalitně pořízeným zvukem z mikrofónů na základě kterých se pak chybně určí TDOA. Pro minimalizaci těchto chyb byl implementován pozicový filtr. Filtr si pamatuje poslední tři pozice a pokud se vzdálenost nově vypočítané hodnoty liší od těžiště těchto zapamatovaných hodnot o více než předem určenou prahovou vzdálenost, která může být dána např. maximální vzdáleností, kterou ujde člověk za daný časový interval, lze tuto vypočítanou polohu považovat za odchylku od normálního chování a může být ignorována. V takovém případě je ale i tato hodnota přidána do historie pro případ, že se v prostoru střídá v konverzaci více lidí. Pokud dojde k takovéto konverzaci, filtr se přeorientuje na nový zdroj zvuku po určité časové době.

4.6 Pomocné grafické rozhraní

K pohodlnější práci s mikrofonním polem bylo vytvořeno pomocné grafické rozhraní (dále jen PGR), kde se v dvourozměrném prostoru vykresluje odhadovaná pozice zdroje zvuku a zobrazují se zde potenciálně užitečné informace pro ladění jako např. časový průběh obdrženého signálu z mikrofonního pole nebo výsledky korelace. Toto rozhraní lze vidět na obrázku 4.1.

Toto rozhraní je napsáno v programovacím jazyce C++ a je vytvořeno v prostředí Borland C++ Builder 5 vytvořené společností Borland.



Obrázek 4.1: Pomocné grafické rozhraní určené k ladění přesnosti používaných algoritmů.

Kapitola 5

Testování

Za účelem testování algoritmu bylo vytvořeno několik metrik, podle kterých lze posoudit celková přesnost mikrofonního pole. Testoval jsem zejména přesnost korelace dvou signálů a schopnost mikrofonního pole určit polohu zdroje zvuku. Určování polohy cíle jsem rozdělil do dvou částí: detekce nehybného zdroje zvuku a schopnost sledovat pohybující se zdroj zvuku.

Testování přesnosti probíhalo s využitím pomocného grafického rozhraní (PGR), kde bylo možné sledovat průběh přijatého signálu, výsledek korelace a jednotlivé odchylky od reálné pozice zdroje zvuku.

5.1 Přesnost korelace

Na přesnosti korelace je závislý celý algoritmus. Pokud udává korelace nepřesné výsledky, algoritmus pro výpočet polohy zdroje zvuku chybně určí polohu. Je tedy zapotřebí aby korelace dávala co nejlepší výsledky.

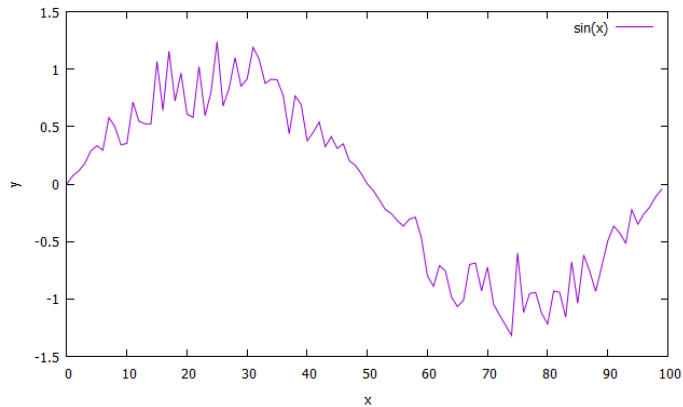
Přesnost korelace jsem testoval následujícím způsobem. Vygeneroval jsem dvojici signálů o jedné periodě funkce sinus délky 100 vzorků. Poté jsem jeden z těchto signálů posunul vůči tomu druhému o náhodnou hodnotu.

Účelem tohoto testování bylo zjistit, jaké procentuální zkreslení signálu ještě dává přesné výsledky. Při posunu signálu jsem vzorky, které dříve nebyly definované, nahradil nulou. Dále jsem každý vzorek tohoto signálu zkreslil náhodnou hodnotou z dříve zvoleného intervalu, kdy bylo cílem zkreslit signál, ale zachovat alespoň přibližný tvar funkce sinus. Jeden takto upravený signál lze vidět na obrázku 5.1.

Poté jsem porovnával původní posun signálu s vypočítaným posuvem pomocí korelace. Odchylky od vypočítaného a reálného šumu a průměrné odchylky v závislosti na míře šumu v signálu lze vidět v tabulce 5.3.

| šum | max. odchylka | prům. odchylka |
|-----|---------------|----------------|
| 1% | 0 | 0 |
| 10% | 1 | 0.003 |
| 25% | 3 | 0.628 |
| 50% | 4 | 0.891 |

Tabulka 5.1: Tabulka popisující přesnost korelace. Obsahuje intenzitu šumu, maximální odchylku a průměrnou odchylku.



Obrázek 5.1: Ukázka deformovaného signálu funkce sinus. Zde byl přidán náhodný šum o maximální hodnotě 0,5 funkční hodnoty v daném bodě. Na ose x je zobrazen časový průběh a na ose y funkční hodnota

První sloupec této tabulky dává maximální hodnotu šumu v poměru k maximální amplitudě signálu, druhý sloupec popisuje maximální odchylku vypočítané pozice od reálného posunutí a ve třetím sloupci je vypočítána průměrná odchylka všech vypočítaných korelací. Z tabulky lze vidět že i obyčejná korelace dává přijatelné výsledky.

5.2 Přesnost algoritmu pro lokalizaci v ideálních podmínkách

Pro lepší představu o přesnosti lokalizace s využitím hyperbol jsem daný algoritmus testoval v ideálních podmínkách. Ideální podmínky zahrnovaly znalost pozic tří mikrofonů a náhodně vygenerovanou pozici zdroje zvuku. Z těchto pozic jsem vypočítal vzdálenost vygenerovaného zdroje zvuku k jednotlivým mikrofonům a z těchto vzdáleností vypočítal rozdíly těchto vzdáleností. Tyto časové rozdíly jsou výsledkem ideálního výpočtu křížové korelace a pomocí těchto dat výpočtu vzdálenosti pomocí přesné znalosti rychlosti zvuku.

S použitím těchto rozdílů vzdáleností jsem analyticky počítal průsečík dvou hyperbol, jejichž rovnice jsem vytvořil z těchto rozdílů vzdáleností. Jádrem tohoto výpočtu tvořily dvě rovnice o dvou neznámých.

Ukázalo se ovšem, že jediné nepřesnosti při výpočtu pozice pomocí tohoto postupu docházelo při finální konverzi vypočítaných souřadnic z datového typu `double` na `int` kvůli zaokrouhlovací chybě. Lze tedy tvrdit, že analytický výpočet dosahuje v ideálních podmínkách 100% úspěšnosti.

Nedostatek této metody je, že se v prostoru vyskytují "hluchá místa", která nelze pokrýt těmito hyperbolami. Pokud vygenerovaný zdroj zvuku leží na přímce, na které leží mikrofony, žádný bod na této přímce nemůže být výsledkem průniku hyperbol, jejichž ohniska jsou tvořeny mikrofony.

Dále jsem testoval přesnost optimalizované metody s využitím hyperbol. Tato metoda využívá předpočítanou tabulku hyperbol a poskytuje vyšší rychlost na úkor přesnosti. Přesnost metody lze vidět v následující tabulce:

| vzdálenost(mm) | prům. odchylka(mm) |
|----------------|--------------------|
| 500 | 21.87 |
| 1000 | 42.23 |
| 1500 | 73.62 |
| 2000 | 128.4 |

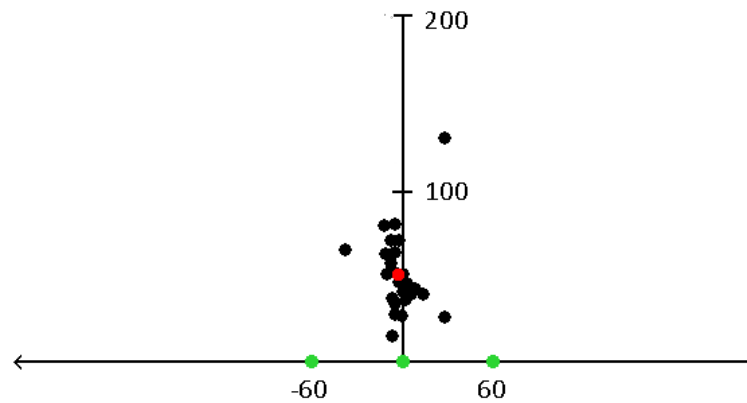
Tabulka 5.2: Tabulka popisující přesnost optimalizované metody hyperbol. Lze vidět, že i v ideálním prostředí při použití této metody dochází k jistým nepřesnostem.

5.3 Lokalizace stacionárního cíle

Při testování lokalizace cíle jsem umístil zdroj zvuku před mikrofonní pole a v PGR sledoval, kde ve 2D ploše určí vypočítanou pozici zdroje zvuku. Vzhledem k tomu že souřadnice 2D plochy lze přepočítat na jednotky vzdálenosti, představoval tento typ testování pomocí PGR efektivní způsob zjištění rozptylu určování pozice okolo reálného zdroje zvuku.

Toto testování probíhalo tak, že jsem mluvil před mikrofonním polem, naměřil svoji pozici vůči mikrofonnímu poli a sledoval pozice, které se postupně vykreslovaly v PGR.

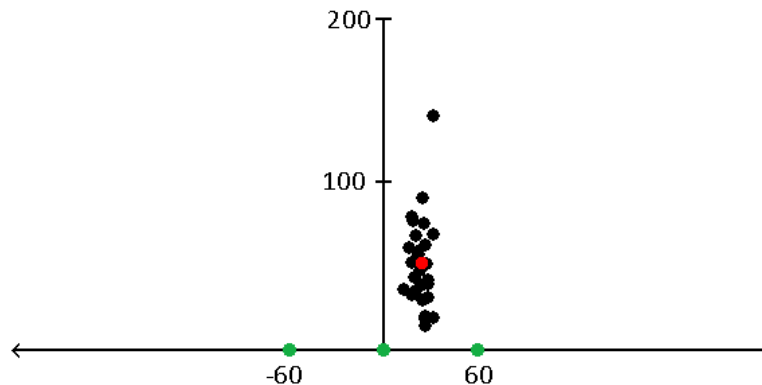
Lokalizace s využitím prototypu mikrofonního pole měla značné odchylky od skutečné pozice zdroje zvuku, ovšem vypočítané body tvořily v okolí skutečné polohy shluk. Vyskytovali se zde i odlehlé hodnoty i přesto, že bylo přidáno filtrování podle pozice. Jeden z výsledků testování lze vidět na obrázku 5.2.



Obrázek 5.2: Rozprostření odhadu vypočítaných pozic mikrofonním polem. Na tomto obrázku lze vidět zeleně vyznačené mikrofony a červeně označenou reálnou polohu zdroje zvuku. Uvedené vzdálenosti jsou v centimetrech.

Lze vidět, že vypočítané polohy se sice pohybují okolo reálné polohy zdroje zvuku, ovšem pro praktické použití je potřeba přesnějšího odhadu. Zpřesnění odhadu polohy bude tedy jedním z hlavních bodů budoucího vývoje.

Výsledek dalšího testu lze vidět na obrázku 5.3. Lze vidět, že vypočítané polohy tvořily podobné shluky jako v prvním případě, ale stejná zůstala i řádová přesnost.



Obrázek 5.3: Rozprostření vypočítaných pozic mikrofonním polem tak, jak byly zobrazeny v pomocném grafickém rozhraní. Zeleně jsou vyznačené mikrofony, červeně je označena reálná poloha zdroje zvuku. Uvedené vzdálenosti jsou v centimetrech.

Lze vidět, že byly testovány pouze lokality v blízkosti mikrofonního pole. Tato vzdálenost byla dána zejména omezením citlivosti použitých mikrofonů. Při větších vzdálenostech nebyl signál tak silný, aby jej zachytila prahová detekce zvuku. I s touto kontrolou vypnutou vypočítalo mikrofonní pole náhodné pozice indikující, že signál se skládal spíše ze šumu než zamýšleného zvuku.

Celkovou přesnost lokalizace lze vidět v následující tabulce:

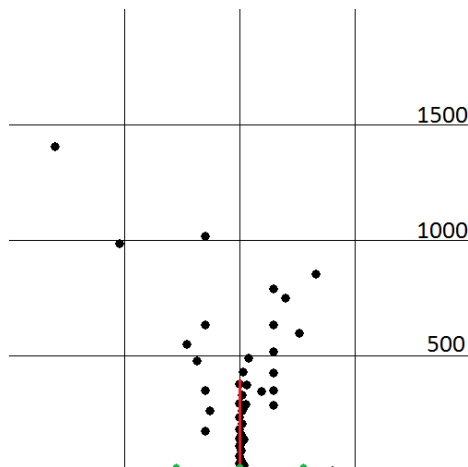
| Vzdálenost(mm) | Úhel($^{\circ}$) | Prům. odchylka(mm) |
|----------------|--------------------|--------------------|
| 500 | 90 | 415.2 |
| 500 | 45 | 448.9 |
| 500 | -45 | 468.1 |

Tabulka 5.3: Tabulka popisující přesnost lokalizace zdroje zvuku v reálném prostředí pomocí prototypu mikrofonního pole. Lze vidět, že přesnost je značně omezena. To je způsobeno zejména malou vzorkovací frekvencí a nekvalitními mikrofony.

5.4 Sledování pohyblivého cíle

Cílem tohoto testu bylo zjistit, do jaké míry je algoritmus schopen sledovat pohybující se cíl. Kvůli omezené přesnosti mikrofonního pole byl testován pouze pohyb po přímce. Při tomto testování jsem nehledal žádné shluky jako tomu bylo u určování pozice stacionárního zdroje. Cílem bylo zejména zjistit, jestli se trajektorie počítaná algoritmem alespoň vzdáleně podobá reálné trajektorii zdroje zvuku.

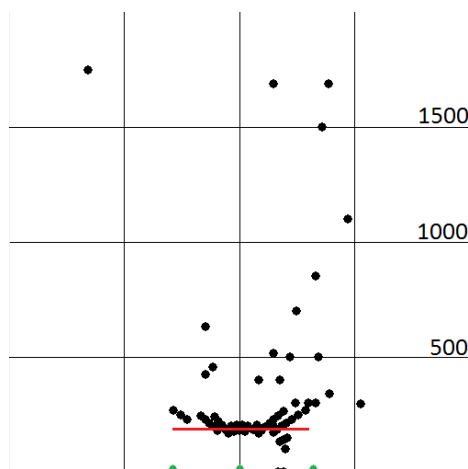
Na obrázku 5.4 lze vidět trajektorii zdroje zvuku, který se pohybuje po přímce kolmé na mikrofonní pole.



Obrázek 5.4: Výsledek sledování pohyblivého cíle pohybujícího se po ose mikrofonního pole. Červeně je vyznačena reálná trajektorie, černě jsou označeny vypočítané pozice a zeleně jsou vyznačeny mikrofony pole.

Z obrázku je zřejmá trajektorie zdroje i přes velké množství odlehlých hodnot.

Na obrázku 5.5 je vidět test sledování cíle, který se pohyboval po přímce rovnoběžné s přímkou kterou tvořilo pole.



Obrázek 5.5: Výsledek sledování trajektorie rovnoběžné s přímkou, kterou tvořily mikrofony pole. Červeně je označena trajektorie zdroje, černě jsou vyznačeny vypočítané pozice a zeleně jsou označeny mikrofony.

Z obrázku je patrné, že sledování této trajektorie dopadlo relativně úspěšně dokud se zdroj příliš nevzdálil od okrajových mikrofonů. Na krajích trajektorie lze vidět, že vzdálenost od protějšího mikrofonu již přesahuje meze pole a mikrofon již není schopen pořizovat kvalitní nahrávky, podle kterých by se dala úspěšně počítat korelace.

Z testování trajektorie jsou jasně zřejmá omezení mikrofonního pole. Pokud se zdroj zvuku přiblížil k okraji mikrofonního pole, jeden mikrofon již nebyl schopen zachytit přesný signál a tím se celý odhad pozice zhoršil.

Celé toto testování sledování pohybujícího se zdroje bylo prováděno pouštěním nahrávky z mobilního zařízení. Obsahem nahrávky byla namísto řeči hudba. Tato skutečnost nejspíše ovlivnila úspěšnosti lokalizace zdroje zvuku.

Kapitola 6

Sestava ARM/SHARC

Tato sestava byla zapůjčena VUT jako přípravek pro implementaci algoritmů pro lokalizaci zdroje zvuku. Sestava je schopna pracovat až s 16-ti mikrofony současně pomocí kterých je schopna provádět lokalizaci cíle.

Sestavu lze vidět na obrázku 6.1.



Obrázek 6.1: Na obrázku lze vidět sestavu ARM pomocí které je prováděna lokalizace.

Z obrázku jsou patrné napájecí sloty, USB porty, ethernetový port a 16 XLR slotů pro konektory mikrofonů.

6.1 Procesor SHARC

SHARC (Super Harvard Architecture Single-chip Computer) je rodina vysoce výkonných 32-bitových floating-point procesorů založených na **super harward architecture** vyráběných firmou Analog Devices, Inc. Jedná se o procesory které jsou uzpůsobené ke zpracování signálů v reálném čase. Poskytují možnost pracovat se signály jak v pevné, tak plovoucí řádové čárce. Zaměřují se zejména na rychlost zpracování a práce se signály.

V SHARC procesorech je možné použít i `Single Instruction Multiple Data` (SIMD) paralelismus dále urychlující operace související se zpracováním signálů. SIMD je typ paralelního zpracování, kdy se jedna operace provede nad více daty současně.

Tyto vlastnosti jej dělají ideálním pro potřeby zpracovávání signálů a lokalizace zdrojů pomocí zachycených signálů.

6.2 Použité mikrofony

K pořizování signálů, podle kterých se počítá poloha zdroje zvuku jsou použity mikrofony `PM0F-6027PN-42UQ`. Jedná se o standardní všesměrové mikrofony schopné nahrávat signály o frekvenci 20 – 16000 Hz. Tyto mikrofony používají fantomové napětí a jsou schopny pořizovat kvalitnější nahrávky než ty, se kterými jsem pracoval u původního prototypu. Mikrofony pořizují analogový signál, který je potřeba nadále zpracovat navzorkováním na digitální hodnoty.

6.3 Práce se sestavou

Na ARM sestavě běží 32-bitový linux operační systém. Celá sestava ovšem nemá žádné porty na vstupní nebo výstupní zařízení, celá práce probíhá přes ethernetový port pomocí SSH komunikace.

Zařízení nemá žádný pevný disk, systém a veškeré soubory potřebné pro běh systému jsou uloženy na odnímatelné paměťové kartě. Operační systém má omezenou funkcionalitu. Není možné např. dlouhodobě ukládat soubory, je tedy potřeba po startu systému nahrát všechny potřebné soubory, nebo vytvořit nový `boot image` operačního systému, na kterém již budou tyto soubory uloženy.

Na sestavě není dostupný překladač pro jazyky C/C++. Programování pro tuto sestavu spočívá v `cross-kompilaci` požadovaného programu na jiném stroji a přehráním přeloženého programu v binární spustitelné podobě na sestavu přes FTP. Protože veškerá práce se sestavou probíhá přes SSH připojení je potřeba k práci další stroj, přes který se budou zobrazovat výsledky a dávat příkazy sestavě.

Toto programování jsem řešil přes virtuální stroj Linux Ubuntu 16.04 UTC, který byl spouštěn v programu `Oracle VM VirtualBox` vyvíjeném firmou Oracle. Na tomto virtuálním stroji probíhala veškerá instalace potřebných knihoven pro běh programu a následný překlad zdrojových kódů. Celý program byl vyvíjen na operačním systému Linux. Vzhledem k tomu, že se jedná o hardwarově specifický program, který používá některé specializované knihovny nebyla v této práci řešena přenositelnost na operační systém Windows.

6.4 Překladač

Protože na sestavě a na mém počítači běželi různé operační systémy, bylo potřeba použít speciální překladač pro `cross-kompilaci`. `Cross-compiler` je překladač, který je schopný přeložit program pro jinou architekturu, než na které se právě nachází. Za tímto účelem jsem použil překladač `arm-linux-gnueabi-hf-g++`. Jedná se o balíček určený ke `cross-kompilaci` na ARM architektury, na kterých běží operační systém Linux. Jedná se o balíček, který je dostupný ze standardních repositářů pro operační systém Linux. Těchto překladačů existuje několik a jsou určené pro specifické ARM desky, na kterých běží operační systém Linux.

6.5 ALSA knihovna

ALSA (Advanced Linux Sound Architecture) je knihovna obsahující API pro pořizování záznamu z mikrofonů operačním systémem linux. Knihovna je napsána v jazyce C a tudíž je vhodnou volbou pro potřeby této práce.

Průběh signálu je mikrofony nahráván analogově a poté je pomocí ADC převodníků převeden na digitální hodnoty požadovaného typu.

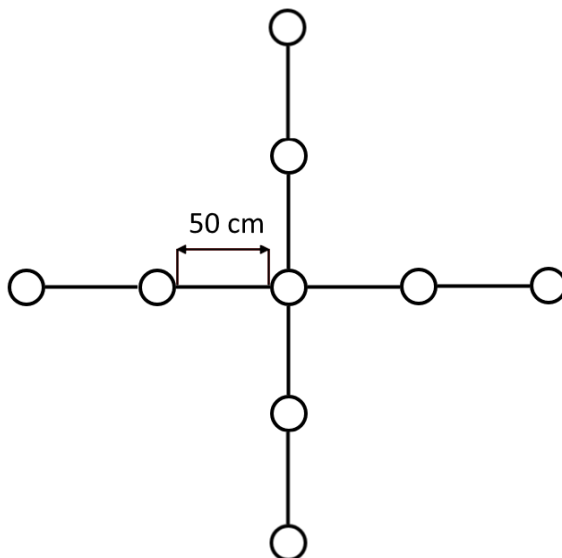
Knihovna poskytuje 2 způsoby čtení hodnot signálu z mikrofonů. Je to `blocking` a `non-blocking`. Rozdíl mezi těmito přístupy je ten, že při čtení v `blocking` módu je běh programu pozastaven, dokud není k dispozici uživatelem požadovaný počet vzorků signálu. Při použití v `non-blocking` módu knihovna vrátí tolik hodnot, kolik je zrovna k dispozici. Při práci s knihovnou jsem používal pouze `blocking` mód abych se vyhnul režii spojené s čekáním na data ze signálu.

Další významější funkcionalitou knihovny je konverze na požadovaný datový typ načtených hodnot. Konverze sice není v základu k dispozici, ale ALSA knihovna ovšem obsahuje vysoce optimalizované pluginy pro konverzi těchto datových typů.

6.6 Struktura mikrofonního pole

Pole je konstruováno podobně jako tomu bylo u prototypu. Mikrofony jsou umístěny v jedné rovině. Mikrofonní pole má tvar kříže stejně jak tomu bylo i u prototypu. Takový tvar má několik výhod. Díky tomuto tvaru je možné lokalizaci ve 3D prostoru rozdělit na dvě podúlohy lokalizace ve 2D prostoru. V jedné úloze se počítá x a y souřadnice a ve druhé se počítá y a z . Tyto souřadnice mají počátek ve středovém mikrophonu. Veškeré určování polohy se počítá vzhledem k současné pozici mikrofonního pole.

Strukturu mikrofonního pole lze vidět na následujícím obrázku.



Obrázek 6.2: Na obrázku lze vidět verzi mikrofonního pole s devíti mikrophony. U této verze je vzdálenost mezi mikrophony 50 cm a celková velikost pole je 2x2 metry.

Alternativou bylo rozmístit mikrofony do mřížky 3x3. Toto rozmístění má ovšem tu nevýhodu, že je potřeba spočítat souřadnice separátně z každého řádku mřížky, tím dostaneme 3 souřadnice, ze kterých je ještě potřeba vypočítat výslednou souřadnici pomocí `time of arrival` metody. Zdroj zvuku má totiž rozdílnou vzdálenost od každého řádku mikrofonů mřížky. vypočítané souřadnice se tedy budou mírně lišit. To znamená že bych musel provádět další výpočet, které může potenciálně přidat další nepřesnosti do výsledného výpočtu pozice.

6.7 Struktura programu

Následující sekce se zabývá strukturou programu. Ta se změnila oproti implementaci vyrobeného prototypu. Program již nemusí posílat data mezi zařízeními, protože vše funguje na ARM architektuře na které se i čtou data z mikrofonů a díky relativně vysokému výkonu procesoru lze provádět veškeré operace přímo na této sestavě.

6.7.1 Čtení dat

Čtení dat probíhá za použití ALSA knihovny. Data se čtou z 9 mikrofonů, pomocí kterých probíhá lokalizace. Na sestavě jsou k dispozici konektory k mikrofonům, ovšem číselná označení těchto konektorů nejsou seřazena. Experimentálně zjištěná číselná označení kanálů lze vidět na následujícím obrázku 6.3.



Obrázek 6.3: Na obrázku lze vidět označení slotů, do kterých se připojují mikrofony na ARM sestavě. Lze vidět, že čísla nejsou seřazena a bylo potřeba experimentálně zjistit, které sloty jsou označeny pod kterými čísly.

Na začátku programu je nutné inicializovat knihovnu s parametry potřebnými pro korektní fungování programu. Mezi tyto parametry patří délka rámců, vzorkovací frekvence, počet kanálů, formát výstupních dat apod.

ALSA knihovna používá pro ukládání dat kruhový `buffer`, do kterého ukládá data asynchronně. Pokud se tedy po nějakou dobu nebudou číst data z bufferu, stará data budou

přepsána novými a stará budou ztracena. Pro správnou funkčnost je tedy nutné periodicky číst data.

Jako základní formát výstupních hodnot poskytuje ALSA knihovna hodnoty ve formátu `float`, ovšem pomocí pluginů lze tyto hodnoty konvertovat. Při práci s daty používám 16-bitový `signed integer`. S podobnými hodnotami jsem pracoval i při práci s prototypem.

6.7.2 Předzpracování dat

Po úspěšném přečtení dat je před samotným výpočtem potřeba data předzpracovat. Toto předzpracování probíhá stejně jako u prototypu. Je potřeba odfiltrvat frekvence, které se nevyskytují v lidské řeči a dále je potřeba provést normalizaci všech kanálů na předem zvolenou střední hodnotu pro jednodušší práci se signálem při korelaci.

Filtrace probíhá pomocí `band-pass` filtru s `cut-off` frekvencemi 80 až 3000 Hz. Tedy veškeré frekvence, které se vyskytují v tomto intervalu budou zachovány a frekvence mimo tyto intervaly budou odstraněny.

Normalizace je prováděna pomocí následujících dvou rovnic:

$$norm = \frac{1}{N} \sum_{n=0}^N f[n]$$
$$f[n] = f[n] * \frac{400}{norm}$$

kde $f[n]$ je zpracovávaný signál, N je délka signálu a 400 je požadovaná střední hodnota. Tato hodnota může být libovolně zvolena tak, aby se s ní dobře počítalo.

Kvůli omezenému výkonu ARM sestavy jsem ovšem nepoužíval plovoucí okno, které jsem používal při práci s prototypem. Toto bylo způsobeno nižším výkonem sestavy oproti PC na kterém běžel původní algoritmus a vyšším počtem mikrofonů, ze kterých bylo potřeba zpracovávat signály. Nahrával jsem celý rámeček najednou s pomocí něj pak prováděl lokalizaci.

Toto plovoucí okno ovšem bylo jedním z hlavních důvodů, kvůli kterému musel algoritmus běžet rychle. Odstraněním tohoto okna jsem si mohl dovolit přidat oproti prototypu další mikrofony a experimentovat se změnou vzorkovací frekvence signálu z 8 kHz na 16 kHz.

Změnou těchto parametrů se celý výpočet zpomalil, ovšem pokud uvážíme, že je celý algoritmus určen k lokalizaci osob, tak je 1 výpočet pozice za vteřinu přijatelná rychlost.

6.7.3 Lokalizace

Lokalizace probíhá stejně jako u prototypu. Dělí se na dvě části: výpočet korelace a výpočet polohy. Tyto algoritmy se od použití u prototypu prakticky nezměnily. Korelace se počítá pomocí klasické varianty korelace (CC). Výsledkem korelace je posunutí signálů v časové rovině, ze kterého se spočítá časový rozdíl a dále rozdíl vzdáleností od jednotlivých mikrofonů. Pomocí těchto rozdílů lze vytvořit hyperboly, na kterých leží zdroj zvuku a z průniku těchto hyperbol lze vypočítat pozici zdroje zvuku.

Průnik hyperbol jsem počítal pomocí optimalizované hyperbolické metody. I přesto, že je analytické řešení přesnější, hluchá místa představovaly problém a optimalizovaná hyperbolická metoda má sice také hluchá místa, ovšem tato hluchá místa se vyskytují na okraji prostoru pokrývaného mikrofonního pole a lze jednodušeji definovat pole působnosti.

Kapitola 7

Výsledky lokalizace zdroje zvuku pomocí ARM sestavy

V rámci této práce jsem testoval několik použití vytvořeného mikrofonního pole. Zejména jsem se zaměřil na lokalizaci statického zdroje zvuku a dále i na sledování pohyblivého cíle. Veškeré testování probíhalo srovnáváním vypočítaných souřadnic s reálnou pozicí zdroje zvuku.

7.1 Lokalizace statického zdroje zvuku

Při testování lokalizace statického zdroje zvuku jsem se zaměřil zejména na přesnost vypočítaných souřadnic. Testování probíhalo naměřením reálných souřadnic vztažených na polohu mikrofonního pole a tyto reálné souřadnice jsem porovnával s vypočítanými.

Pro zvýšení přesnosti probíhalo testování pomocí mobilního reproduktoru, na kterém běžela nahrávka lidské řeči. Tento přístup jsem použil protože bylo jednodušší měřit reálnou polohu malého zařízení než lidského řečníka.

Pro lepší představu jsem testoval i lokalizaci mluveného projevu lidskou osobou, abych zjistil, zda má vůbec cenu provádět testování pomocí mobilního reproduktoru. Testoval jsem pouze lokalizaci řečníka na jedné poloze a výsledky jsem porovnával s výsledky lokalizace mobilního reproduktoru. Tyto výsledky lze vidět v následující tabulce.

Tabulka 7.1: Měření odchylky mezi lokalizací mobilního zařízení a lidské osoby.

| Průměrná odchylka lok. lidské osoby (mm) | Průměrná odchylka lok. mob. zařízení (mm) |
|------------------------------------------|-------------------------------------------|
| 203.37 | 162.89 |

Rozdíl mezi lidským řečníkem a mobilním reproduktorem byl asi 4 cm, což je dostatečně malý rozdíl.

Výsledky pořízené pomocí mobilního reproduktoru je možné vidět v tabulce 7.2. Testy jsem dělil podle vzdálenosti od mikrofonního pole, ovšem bral jsem v úvahu i úhel, pod kterým dopadal zvuk na mikrofonní pole. Úhlem je myšlena přímka, která vznikne pomyslným spojením pozice zdroje zvuku se středovým mikrofonem.

Tabulka 7.2: Měření přesnosti lokalizace na základě polohy vůči mikrofonnímu poli.

| Vzdálenost od mikr. pole (mm) | Úhel(°) | Průměrná odchylka (mm) |
|-------------------------------|---------|------------------------|
| 250 | 45 | 204 |
| 250 | 90 | 162 |
| 500 | 45 | 368 |
| 500 | 90 | 167 |
| 750 | 45 | 548 |
| 750 | 90 | 280 |

Tyto testy probíhaly s vzorkovací frekvencí 8 kHz a délkou nahraného zvuku 0,5 sekundy. Testování probíhalo v místnosti, ve které bylo kromě řečníka ticho a tedy minimální šum.

7.1.1 Zhodnocení úspěšnosti

Z tabulky lze vidět, že co se týče přesnosti algoritmu pro lokalizaci, tuto přesnost je ještě možné dále vylepšovat. Celková odchylka, kterou si může algoritmus dovolit, záleží na použití. Vzhledem k tomu, že algoritmus je určen pro studijní účely, dala by se výsledná přesnost označit za dostačující uvažíme-li velikost cíle, který se snažíme lokalizovat, což je osoba. Pokud ovšem uvažíme potřebu lokalizace vysoké přesnosti, výsledná přesnost algoritmu není pro tyto potřeby dostačující.

Přesnost je také velmi závislá na vzdálenosti od mikrofonního pole. Pokud se zdroj zvuku nachází na pozici, která je od mikrofonního pole příliš daleko, mikrofony jej "neuslyší" a lokalizace selže. Toto lze vidět u pozic, kdy byl úhel pod kterým dopadal signál na mikrofonní pole 45°, kdy jeden mikrofon měl problémy zaslechnout řečníka.

Pokud ovšem zanedbáme případné nepřesnosti v lokalizaci, dalším problémem, který má toto mikrofonní pole, je velikost pokrytí prostoru. Pokud máme omezený počet mikrofonů je nutné zvolit kompromis mezi přesností lokalizace a velikostí prostoru, který je schopné mikrofonní pole pokrýt. Uvažíme-li velikost některých místností (např. přednášková místnost D105), toto mikrofonní pole není schopné pokrýt ani 1/4 z celkové velikosti. Závěr, který je možné z těchto faktů vyvodit je, že pro přesnou lokalizaci velkého prostoru je nutný daleko větší počet mikrofonů, než 9 se kterými jsem pracoval popř. 16 což je maximum se kterými je schopná ARM sestava pracovat. Přidáním dalších mikrofonů ovšem roste potřeba vyššího výkonu pro zpracování dat z těchto mikrofonů.

Asi největším problémem na který jsem ovšem narazil je výkon ARM sestavy, na které běží algoritmus. Pro počítání v reálném čase bylo potřeba používat nižší vzorkovací frekvence než jakých lze pomocí ALSA knihovny dosáhnout. Experimentoval jsem s frekvencemi 8 a 16 kHz. Také bylo nutné zredukovat délku nahrávky signálu se kterou jsem pracoval.

Pokud ovšem chceme zpracovávat nahrávky s vysokou vzorkovací frekvencí (48 kHz) z většího počtu mikrofonů a počítat korelaci na signálech s delším časovým rozsahem, je potřeba mnohem větší výkon, než kterého je ARM sestava schopná.

7.2 Lokalizace pohyblivého zdroje zvuku

Na lokalizaci pohyblivého zdroje zvuku jsem nekladl velký důraz. Zejména proto, že algoritmus nepoužívá speciální heuristiky pro lokalizaci pohyblivého cíle jako např. **movement prediction**, která by mohla v teorii zvýšit přesnost lokalizace při pohybu. Sledování pohyblivého cíle se skládá z bodů, na kterých se zdroj zvuku nacházel v průběhu pohybu.

Dá se tedy očekávat, že má lokalizace pohyblivého zdroje zvuku podobnou přesnost jako lokalizace nehybného zdroje. Místo experimentů s pohyblivým cílem jsem se zaměřil spíše na experimenty se změnami parametrů při výpočtu.

7.3 Změny parametrů pro lokalizaci

Jak bylo uvedeno dříve přesnost lokalizace je závislá na několika parametrech. Těmito parametry jsou: vzorkovací frekvence a délka rámce se kterým se pracuje. Testoval jsem nejen do jaké míry ovlivňují tyto parametry přesnost celkového algoritmu ale také do jaké míry tyto parametry ovlivňují rychlost celého výpočtu. V následující tabulce jsou zobrazeny výsledky testů různých vzorkovacích frekvencí a do jaké míry ovlivňují přesnost lokalizace.

Veškeré testy probíhali ve vzdálenosti 750 cm od pole s mobilním reproduktorem umístěným přímo před středovým mikrofonem.

Tabulka 7.3: V tabulce lze vidět závislost přesnosti na vzorkovací frekvenci přijatého signálu.

| Vzorkovací frekvence(kHz) | Průměrná chyba lokalizace(mm) |
|---------------------------|-------------------------------|
| 8 | 171 |
| 16 | 109 |
| 24 | 97 |
| 32 | 96 |
| 48 | 88 |

Při testování jsem používal takovou velikost rámce, aby se do něj vešla vteřina nahrávky z mikrofonů. Z tabulky lze vidět, že zvýšením vzorkovací frekvence lze zvýšit přesnost celé lokalizace. Zvýšení přesnosti ovšem není tak velké, aby se tím dala ospravedlnit zvýšená doba výpočtu.

Problém s touto přesností je ten, že při takových velikostech dat, se kterými jsem zde pracoval, již není tento algoritmus schopen dostatečně rychle počítat pozice.

Testoval jsem i rychlost celého algoritmu při použití různých délkách rámců. Délka rámce je závislá na délce nahrávaného signálu a na vzorkovací frekvenci. Při tomto testu jsem zvolil délku nahrávaného signálu jednu vteřinu a měnil vzorkovací frekvence a sledoval rychlost výpočtu.

Výsledky rychlosti výpočtu lze vidět v následující tabulce. Kvůli dlouhé době výpočtu toto testování probíhalo pouze se třemi mikrofony a na přesnost výpočtu jsem nebral ohled.

Tabulka 7.4: Délka výpočtu algoritmu při různých délkách rámců.

| Délka rámce | Délka výpočtu(s) |
|-------------|------------------|
| 8000 | 2.14 |
| 16000 | 4.52 |
| 24000 | 7.05 |
| 32000 | 9.83 |
| 48000 | 16.92 |

Z tabulky lze vidět, že pro rozumnou rychlost celkového výpočtu je nutné zvolit kompromis mezi přesností a dobou výpočtu, protože ARM sestava má oproti počítači, na kterém běžel algoritmus při pokusech s prototypem, velice omezený výkon.

7.3.1 Shrnutí experimentů

Z experimentů s ARM sestavou je vidět, že při reálném použití je nutné limitovat délku rámce se kterým se pracuje. V tabulce 7.4 je vidět, že pokud bychom chtěli pracovat s opravdu dlouhými rámci, je nutné zvýšit výkon sestavy, na které bude běžet tento algoritmus.

Při použití této sestavy je nutné zvolit kompromis mezi délkou rámce, vzorkovací frekvencí a případně počtem použitých mikrofonů. Tím se sice sníží celková přesnost lokalizace, ale dosáhneme alespoň rychlosti jeden výpočet pozice za jednotky vteřin.

Kapitola 8

Závěr

V zimním semestru jsem sestavil jednoduché mikrofonní pole pomocí kterého jsem se seznámil s problematikou lokalizace zdroje zvuku, možnostmi jaké v této oblasti existují a algoritmy pomocí kterých lze tyto problémy řešit. Nastudoval a implementoval jsem algoritmus pro výpočet TDOA. Jako algoritmus pro tento výpočet jsem zvolil křížovou korelaci nahraných signálů. Dále jsem se pokusil na základě tohoto algoritmu odhadnout polohu zdroje zvuku. Polohu zdroje zvuku jsem počítal pomocí hyperbolické metody a tuto metodu i implementoval. Poloha je vypočítána v kartézském souřadném systému a lze ji jednoduše transformovat i do polárních souřadnic. Dále jsem otestoval přesnost těchto algoritmů na reálných datech.

V letním semestru jsem obdržel ARM sestavu, která tvoří jádro celého mikrofonního pole. Dále jsem upravil algoritmy použité při konstrukci prototypu mikrofonního pole tak, aby byly schopné pracovat na ARM sestavě, která má omezený výkon. Na ARM jsem nainstaloval mikrofony a oproti prototypu přidal více mikrofonů. Kvůli většímu počtu mikrofonů a omezenému výkonu jsem byl nucen upustit od použití klouzavého okna a pracovat s přímo nahranými hodnotami z mikrofonů. Po modifikaci původního algoritmu jsem otestoval přesnost celého mikrofonního pole.

Přesnost lokalizace zdroje zvuku pomocí mikrofonního pole je pro praktické účely nutno ještě vylepšit ovšem jako výsledek pokusů s tímto mikrofonním polem se dá označit za dostačující. Přesnost je možné nadále vylepšit použitím kvalitnějších mikrofonů nebo použitím jiných metod pro korelaci signálů.

8.1 Přesný výpočet pozice

Největším problémem při výpočtu pozice zdroje zvuku je omezený výkon celé soustavy. Pokud bychom chtěli provádět opravdu přesnou lokalizaci zdroje zvuku tedy pracovat alespoň se vzorkovací frekvencí 32 nebo 48 kHz, přitom pracovat s rozumně dlouhou nahrávkou zdroje zvuku (alespoň 1 sekunda) a zpracovávat nahrávky z většího mikrofonního pole, je výkon této soustavy nedostatečný. Už při pokusech se třemi mikrofony, vzorkovací frekvencí 32 kHz a nahrávce jedné sekundy tj. (32000 hodnot pro korelaci) přesáhla doba běhu výpočtu několik vteřin, což je pro praktické použití nevyhovující.

8.2 Možnosti dalšího vývoje

Jako jedna z možností vylepšení tohoto mikrofonního pole je použití mikrofonů s větším dosahem popř. mikrofony, které jsou schopny pořídit kvalitnější nahrávky. Pokud uvažujeme změny týkající se mikrofonů, další možností je použití více mikrofonů. Touto změnou je teoreticky možné zlepšit přesnost a zvětší se prostor, který je schopno toto pole pokrýt. Toto vylepšení by ovšem vyžadovalo změnu architektury, na které by tento algoritmus běžel, protože ARM sestava má omezený počet mikrofonů se kterými může pracovat současně a nedostatečný výkon.

Další možností použití jiných algoritmů pro lokalizaci. Zejména algoritmy, které provádějí přesnější korelaci signálů. Použití korelačních algoritmů, které se lépe vypořádají s případným šumem je teoreticky možné dále zlepšit přesnost celé lokalizace. To samé platí i pro část algoritmu provádějící samotnou lokalizaci pomocí TDOA.

Poslední možností jak vylepšit toto pole je přidání přívětivějšího uživatelského rozhraní. Prozatím je výstupem programu bod v prostoru, který vydává zvuk. Toto je možné například přidáním grafického uživatelského rozhraní nebo přidáním hardwarového ukazovátka, které se bude pomocí serv otáčet za zdrojem zvuku.

Literatura

- [1] Benesty, J.; Chen, J.; Huang, Y.: *Microphone Array Signal Processing*. Springer Topics in Signal Processing, Springer Berlin Heidelberg, 2008, ISBN 9783540786122.
- [2] Cengiz, M.; Kubilay, I. A.; Birant, D.: Emotion Extraction from Human Voice Using Voice Frequency Spectrum Analysis. Technická zpráva.
- [3] Chan, Y. T.; Ho, K. C.: A Simple and Efficient Estimator for Hyperbolic Location. *IEEE Transactions on Signal Processing*, ročník 42, č. 8, 1994, ISSN 1053-587X.
- [4] Cramer, O.: The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and CO₂ concentration. *The Journal of the Acoustical Society of America*, ročník 93, č. 5, 1993: s. 2510–2516.
- [5] Heiß, H.; Pepper, P.; Schlingloff, H.; aj. (editoři): *Determining Times of Arrival of Transponder Signals in a Sensor Network using GPS Time Synchronization*, LNI, ročník 192, GI, 2011, ISBN 978-88579-286-4-7.
- [6] Huang, K. N.; Huang, C. F.; Li, Y. C.; aj.: High precision, fast ultrasonic thermometer based on measurement of the speed of sound in air. *Review of Scientific Instruments*, ročník 73, č. 11, 2002: s. 4022–4027.
- [7] Nguyen, U. T. K.; Pham, T. V.: Performance assessment of generalized cross-correlation based algorithms for multisource point-based localization and detection. In *The 2011 International Conference on Advanced Technologies for Communications (ATC 2011)*, Srpen 2011, ISSN 2162-1039, s. 303–306.
- [8] O’Keefe, B.: Finding Location with Time of Arrival and Time Difference of Arrival Techniques. Technická zpráva, ECE Senior Capstone Project, 2017.
- [9] Parham, A.: The Fusion of Distributed Microphone Arrays for Sound Localization. *EURASIP Journal on Applied Signal Processing*, ročník 4, 2003: s. 338–347.
- [10] Schultheiss, P. M.; Wagner, K.: *Active and Passive Localization: Similarities and Differences*. Dordrecht: Springer Netherlands, 1989, ISBN 978-94-009-2289-1, s. 215–232.
- [11] Smith, S.: *The Scientist and engineer’s guide to digital signal processing*. California Technical Publishing, 2002, ISBN 0966017633.
- [12] Ward, M. B. D.: *Microphone Arrays, Signal Processing Techniques and Applications*. Springer-Verlag Berlin Heidelberg, první vydání, 2001.

Příloha A

Plakát

Součástí zadání bylo vytvořit plakát velikosti A2. Plakát lze vidět na následujícím obrázku.

The poster features a blue header with the Faculty of Informatics and Technology logo (a red square with a white 'T') and the text 'FAKULTA INFORMAČNÍCH TECHNOLOGIÍ'. To the right, it lists the author 'Autor: Bc. Zdeněk Pelz' and supervisor 'Vedoucí: Ing. Igor Szöke, Ph.D.'.

The main title is 'Akustická detekce pozice řečníka pomocí mikrofonního pole a její vizualizace v prostoru'.

The 'Postup výpočtu:' section includes:

- Nahrání zvuku pomocí mikrofonního pole
- Zpracování signálů a vypočítání jejich zpoždění t_0 a t_1
- Vypočet souřadnic pomocí těchto zpoždění a rovnic pro hyperboly

The hyperbola equation is given as $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$.

The final step is '- Zobrazení souřadnic na počítači'.

The diagram shows a speaker at position $[x;y]$ emitting sound towards a microphone array (Mikr. pole) with two microphones at t_0 and t_1 . The array is connected to an ARM processor, which outputs coordinates $[x;y]$. The ARM is also connected to a PC, which displays the coordinates $[x;y]$ on its screen.

The 'Použité algoritmy:' section lists:

- Křížová korelace
- Band-pass filtr
- Hyperbolická metoda určení polohy

Obrázek A.1: Plakát.

Příloha B

Obsah paměťového média

Na přiloženém CD jsou veškeré zdrojové kódy, které jsem použil k vypracování této práce. Jsou zde zdrojové kódy algoritmu pro lokalizaci, zdrojový kód pro desku v Arduino prostředí a jiné programy pomocí kterých probíhalo testování. Tyto kódy jsou ve složce `src` kde jsou rozděleny do podsložek. Celková struktura obsahu paměťového média je podrobněji popsána v souboru `README`.