

IMPLEMENTACE VIRTUÁLNÍHO AKUSTICKÉHO ZDROJE NA DSP ZA PODPORY MATLABU

F. Rund, M. Švejda

Katedra radioelektroniky, fakulta elektrotechnická,
České vysoké učení technické v Praze, Česká republika

Abstrakt

Vytvoření virtuálního akustického prostoru má svůj význam pro celou řadu aplikací. K nejpoužívanějším způsobům pro vytvoření iluze zvuku přicházejícího z určitého směru patří využití konvoluce vstupního zvuku s HRIR (Head Related Impulse Response – tj. impulzní odezva vztažená k hlavě). Tento článek prezentuje implementaci zmíněného systému v reálném čase, ve které je možno měnit parametry (polohu) virtuálního zdroje, využívající DSP systém Soundart Chameleon a prostředí Matlab. Pro prostředí Matlab je využíváno nejen pro editaci naměřených impulzních odezev a vytvoření HRIR sady v příslušném formátu, ale také pro automatické generování souborů potřebných pro překlad assembleru pro DSP. Pro změnu banky filtrů proto není vyžadována znalost programování DSP, stačí jen přeložit kód připravený Matlabem a načíst jej do DSP.

1 Úvod

Virtuální akustický prostor (Virtual Acoustic Space – VAS) lze využít pro celou řadu aplikací, od zábavního průmyslu (počítačové hry) až po aplikace asistivního charakteru (navigace nevidomých). K nejpoužívanějším způsobům pro vytvoření iluze zvuku přicházejícího z určitého směru patří využití konvoluce vstupního zvuku s HRIR (Head Related Impulse Response – tj. impulzní odezva vztažená k hlavě). Tato impulzní odezva v sobě zahrnuje většinu informací, podle kterých je člověk schopen poznat směr přicházejícího zvuku (časové a intenzitní rozdíly mezi ušima, filtraci odrazy od torza, ramen, hlavy, boltce...).

Zmíněnou sadu HRIR lze získat měřením (podrobnosti např. v [1]), popř. jinými metodami [2]. Virtuální akustický prostor vytvořený za pomoci HRIR lze samozřejmě implementovat i ověřit přímo v prostředí Matlab, jak jsme ukázali v našich předchozích článcích [3,4]. Pro využití virtuálních akustických zdrojů pro asistivní aplikace (např. pro navigaci nevidomých) je ale důležitá implementace zmíněného systému v reálném čase, aby bylo možno měnit parametry (polohu) virtuálního zdroje. Člověk při pokusu lokalizovat zdroj zvuku nevyužívá jen informace přítomné v HRIR, ale důležitá je i vizuální kontrola (u asistivních pomůcek pro nevidomé se pochopitelně neuplatňuje) a (nevědomé) pohyby hlavy. Vlivem pohybů hlavy na přesnost lokalizace zdroje zvuku jsme se již zabývali v našem článku [5], kdy jsme opět využívali prostředí Matlab.

Pro zmíněné účely – ověření chování VAS v reálném čase, včetně zahrnutí vlivu pohybů hlavy – byla vytvořena implementace na systému s digitálním signálovým procesorem, podrobněji popsaná v [6]. Tento článek se zabývá využitím Matlabu při vývoji i provozu zmíněné aplikace.



Obrázek 1: VAS implementovaný na DSP zařízení Soundart Chameleon.

2 Návrh systému

Pro zmíněné ověření VAS vytvořeného prostřednictvím HRIR byl na našem pracovišti implementován experimentální systém, využívající DSP zařízení Soundart Chameleon [7], obr. 1. Jádrem zařízení Chameleon je signálový procesor Freescale 56303, který je nutno programovat v příslušném jazyce symbolických adres (assembleru). Dále toto zařízení disponuje řídicím procesorem Freescale Coldfire, který lze programovat v jazyce C, stereofonním vstupem a výstupem (vzorkovací kmitočet 48 kHz) a MIDI rozhraním. Se zařízením je dodáváno vývojové prostředí Chameleon Development Environment (CDE) a další programové vybavení umožňující nahrávání programů do zařízení. Další podrobnosti lze nalézt v dokumentaci zařízení [7].

Návrhu a implementaci VAS na zařízení Chameleon je věnován článek [6]. Realizovaný systém umožňuje v reálném čase vytvořit ve sluchátkách posluchače dojem umístění zvukového signálu přicházejícího na vstup zařízení do polohy (azimut, elevace) zadané uživatelem. Systém umožňuje zadávat a plynule měnit polohu virtuálního zdroje zvuku buď pomocí ovládacích prvků, nebo vzdáleně, přes protokol MIDI. Systém uvažuje i korekci polohy zdroje podle pohybů hlavy uživatele (při připojení zařízení pro sledování polohy hlavy). Současně s implementací tohoto zařízení byl vyvinut systém pro sledování polohy hlavy uživatele (obrázek 2), který komunikuje s vlastním zařízením pomocí protokolu MIDI.

Vzhledem k tomu, že popisovaná implementace slouží k testování algoritmů pro vytváření VAS, tak obsahuje, kromě implementace přímé konvoluce vstupního signálu s HRIR, i implementaci využívající diferenciální HRIR (DHRIR). Využití DHRIR je nová metoda, publikovaná v [10], která klade menší nároky (zejména z hlediska využití paměti) na výpočetní výkon. Dále implementace umožňuje i srovnání s klasickou „amplitudovou panorámou“ (tj. prosté řízení úrovně levého a pravého kanálu). V případě využití HRIR i DHRIR jsou příslušné koeficienty uloženy v paměti zařízení ve vhodném rozlišení a program jen vybere příslušný „řádek tabulky“ odpovídající požadovanému směru a spočítá konvoluci se vstupním signálem. Vzorčky HRIR jsou tedy přímo součástí kódu pro signálový procesor, proto při výměně sady HRIR je nutný zásah do kódu programu.

Další kapitoly prezentují řešení problému výměny HRIR sady, s využitím prostředí Matlab. Prostředí Matlab je využíváno pro editaci naměřených impulzních odezev a vytvoření HRIR sady v příslušném formátu. Byla vytvořena sada skriptů, které načtou naměřené HRIR, převedou je do správného formátu a automaticky vygenerují soubory potřebné pro překlad assembleru pro DSP. Pro změnu banky filtrů proto není vyžadována znalost programování DSP, stačí jen přeložit kód připravený Matlabem a načíst jej do DSP.



Obrázek 2: Test systému s headtrackingem.

3 Matlab podpora

Jak již bylo zmíněno, systém je navržen tak, že HRIR koeficienty jsou součástí kódu, tj. pro jejich výměnu je nutné znovu přeložit kód a nahrát jej do procesoru. Tento problém by bylo možné řešit např. nahráním nových koeficientů přes MIDI rozhraní, což by ale znamenalo zásadní změnu již implementovaného programu. Proto bylo navrženo dále popsané řešení zmíněného problému s využitím prostředí Matlab. Programy pro Chameleon lze vytvářet ve vývojovém prostředí CDE - Chameleon Development Environment (freeware). Pro vytvoření kompletního funkčního programu pro zařízení Chameleon je třeba vytvořit v CDE projekt. Projekt obsahuje program pro mikrokontroler ColdFire napsaný v jazyce C nebo C++ a program pro signálový procesor napsaný v jazyce symbolických adres (Assembleru) pro DSP 56303. Když jsou připraveny programy pro ColdFire i DSP56303, lze projekt zkompileovat a výsledný program nahrát do zařízení. Další podrobnosti lze nalézt v dokumentaci [7]. Zmíněné řešení spočívá ve využití prostředí Matlab pro automatické generování části kódu, obsahujícího novou HRIR banku. Tímto způsobem lze změnit nejen vlastní HRIR (tj. přizpůsobit generovaný VAS danému uživateli), ale také změnit počet použitých koeficientů i krok v azimutu i elevaci. Prostředí Matlab je rovněž využito pro přípravu vstupních HRIR dat.

3.1 Pomocné skripty pro přípravu banky (D)HRIR

Pro úpravu vstupních HRIR dat je připravena následující sada Matlab funkcí a skriptů. Tato sada funkcí zahrnuje převedení HRIR z jiných formátů, a nutné úpravy.

První užitečnou funkcí je funkce `loadstruct.m`, která umožňuje načíst data z formátu IRCAM LISTEN [8] do maticového formátu, který je využíván dalšími funkcemi. Pokud je třeba pracovat s daty v jiném formátu z jiných databází, jsou k dispozici také příslušné skripty pro převod z jiných databází. Funkce je deklarována takto:

```
function [irs,fs]=loadstruct(irs_structL,irs_structR,samples).
```

Vstupními parametry funkce jsou koeficienty ve formátu listen (`irs_structL`, `irs_structR`) a počet vzorků (`samples`), který mají mít výstupní HRIR v maticovém formátu používaném v této implementaci. Některé databáze HRIR nemají mezi sousedními pozicemi (azimut i elevace) lineární rozestupy, pro takový případ je nutné provést interpolaci dat takovým způsobem, aby výsledkem byl uniformní krok v azimutu i elevaci. Celá aplikace v současném stavu je uzpůsobena pouze pro lineární rozestupy mezi sousedními pozicemi.

Skript `loadstruct.m` kromě načtení koeficientů do matice provádí automaticky několik dalších operací. Vzhledem k tomu, že pro efektivní implementaci na DSP je ideální aby počet vzorků HRIR byl 100, je nutné impulzní odezvy nejprve přizpůsobit, aby např. nedošlo k tomu, že u některých odezvy v prvních 100 vzorcích ještě nebude důležitá informace. Tento případ je ošetřen tak, že skript nejprve hledá maxima ve všech odezvěch a jejich polohy. Určí maximum, jehož poloha je nejméně vzdálena od začátku odezvy. Od této polohy odečte konstantu (např. 10) čímž zajistí, že v odezvě bude zachycena důležitá náběžná část této odezvy. Ze všech odezvy jsou vybrány hodnoty až od této polohy. Všechny impulzní odezvy jsou také vynásobeny oknem "Tukey", které zajistí pozvolné utlumení odezvy, aby vždy končily v nule. Dále jsou všechny odezvy normované maximální absolutní hodnotou ze všech odezvy tak, aby byla maximální absolutní hodnota ve všech odezvěch 1.

Další dvě jednoduché funkce slouží k orientačnímu ověření banky HRIR poslechem.

```
function [out_sound]=HRIR_test(irs,fs)
function [out_sound]=DHRIR_test(DHRIR,fs)
```

Vstupními parametry funkce `HRIR_test` je banka HRIR a vzorkovací frekvence (`fs`), na které chceme výstupní zvuk poslouchat. Jako základní zvuk je použit vygenerovaný modulovaný šum. Tento zvuk lze samozřejmě snadno nahradit jakýmkoliv jiným. Skript `DHRIR_test` pracuje podobně s rozdílem, že vstupem je banka DHRIR koeficientů.

Vzhledem k tomu, že systém Soundart Chameleon pracuje se vzorkovacím kmitočtem 48 kHz, je nutné používat HRIR se stejným vzorkovacím kmitočtem. Pokud máme k dispozici banku HRIR na jiném vzorkovacím kmitočtu `fs_old`, lze provést převzorkování na aktuální vzorkovací kmitočet `fs_new=48 000` Hz prostřednictvím funkce s následující deklarací

```
function [irs_resampled] = irs_resample(irs,fs_old,fs_new)
```

Zvolená implementace v zařízení Chameleon využívá banky HRIR s uniformním krokem v azimutu (i elevaci). Vzhledem k použité implementaci se doporučuje použít krok v horizontální rovině (azimut) cca 3 °. Pokud použitá banka HRIR tímto krokem nedisponuje, je možno použít následující funkci

```
function  
[irs_interpolated]=d_interp(irs,stepH_old,stepH_new,stepV_old,stepV_new,Vmin,Vmax.)
```

Vstupem této funkce je banka HRIR v maticovém formátu, původní horizontální krok, nově požadovaný horizontální krok, původní vertikální krok, požadovaný nový vertikální krok, minimální hodnota elevace a maximální hodnota elevace. Výstupem je nová banka HRIR v maticovém tvaru s požadovaným počtem azimutů a elevací (je provedena lineární interpolace). Je samozřejmě možné použít jiný způsob interpolace než lineární, podrobněji se touto problematikou zabývá práce [9].

Pokud je při tvorbě VAS využito diferenciální HRIR, je potřeba použít (analogicky) následující funkci

```
function [irs_interpolatedH]=d_interpHD(irs,step_old,step_new)
```

Vstupními parametry této funkce jsou koeficienty HRIR v horizontální rovině s elevací 0°, původní rozestup mezi sousedními azimuty a požadovaný rozestup. Funkce vytvoří banku DHRIR, pokud je nový horizontální krok menší, tak pozice lineárně proloží. Správnost vygenerované banky DHRIR lze otestovat poslechem pomocí již zmíněné funkce `DHRIR_test.m`.

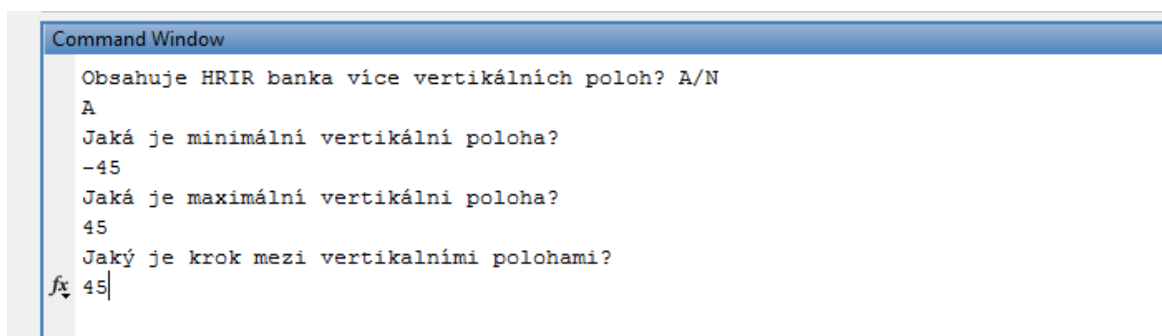
3.2 Skript pro vytvoření projektu s novými HRIR

Pokud jsou připraveny koeficienty v příslušném maticovém tvaru, je možné spustit hlavní skript `HRIR_change` který slouží k přípravě projektu pro Chameleon Development Enviroment s novými koeficienty. Po spuštění tohoto skriptu Matlab vyvolá dialogové okno, které vyzve k zadání složky s m-soubory pro práci s HRIR.

Následně program otevře dialogové okno pro výběr souboru `*.mat`, ve kterém je uložena banka HRIR v maticovém tvaru. Poté se program prostřednictvím příkazové řádky dotáže, zda banka HRIR obsahuje více elevací. Pokud ano, zeptá se dále na minimální a maximální elevaci a krok mezi nimi.

Poté se program zeptá, zda bude požadováno polohování DHRIR. Pokud ano, dojde k vyvolání dialogového okna pro zadání banky DHRIR ve formátu `*.mat`. Dále jsou do příslušné složky vytvořeny soubory, které obsahují příslušnou banku HRIR. Pro mikrokontroler ColdFire je to soubor `diff_hrir_flag.h`. Pro signálový procesor jsou to soubory `define_storage.asm`, `DHRIR_koef_pointery.asm`, `DHRIR_koeficienty.asm`, `HRIR_koef_pointery.asm`, `HRIR_koeficienty.asm`, které jsou vloženy do podsložky `dsp`.

Když jsou soubory úspěšně vytvořeny, je již možné otevřít vývojové prostředí Chameleon Development Environment, otevřít příslušný projekt a přeložit jej obvyklým způsobem.



Obrázek 3. Program zjišťuje parametry banky HRIR prostřednictvím příkazové řádky Matlabu

4 Závěr

V článku byl prezentován komplexní systém využívající DSP zařízení Soundart Chameleon pro vytvoření virtuálního akustického zdroje, jehož parametry (azimut, elevace) lze měnit v reálném čase. Tento článek byl zejména zaměřen na využití prostředí Matlab jako podpory pro generování programu pro zmíněné zařízení. Tímto způsobem je připravena HRIR banka (včetně případné změny rozlišení, interpolace, atp.) a vygenerovány potřebné soubory umožňující načtení této HRIR banky do paměti zmíněného zařízení. Popsaný systém je využíván pro vývoj i testování metod pro virtuální polohování zdrojů zvuku.

Poděkování

Tato práce byla podpořena grantem Studentské grantové soutěže ČVUT č. SGS11/159/OHK3/3T/13.

Reference

- [1] Štorek D. – Suchan, R. Analysis of the Problems in HRTF Measuring, In *POSTER 2011 - 15th International Student Conference on Electrical Engineering*. Prague: CTU, Faculty of Electrical Engineering, 2011, p. 1-4. ISBN 978-80-01-04806-1.
- [2] Rund, F. - Saturka, F.: Alternatives to HRTF Measurement. In *35th International Conference Telecommunications and Signal Processing*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012, p. 648-652. ISBN 978-1-4673-1118-2.
- [3] Rund, F. - Saturka, F. GUI pro syntézu HRTF a testování výsledků. In *20th Annual Conference Proceeding's Technical Computing Bratislava 2012*. Praha: Humusoft, 2012, s. 1-6. ISBN 978-80-970519-4-5.
- [4] Rund, F. Audio Pointer V2 - Joystick & Camera in MATLAB. In: *19th Annual Conference Proceedings Technical Computing Prague 2011*. Praha: Vydavatelství VŠCHT Praha, 2011, p. 1-5. ISBN 978-80-7080-794-1.
- [5] Rund, F. - Štorek, D. Head-Tracking for Virtually Auditory Space Implemented in MATLAB In *20th Annual Conference Proceeding's Technical Computing Bratislava 2012*. Prague: HUMUSOFT, 2012, p. 1-4. ISBN 978-80-970519-4-5.
- [6] Švejda, M. DSP implementation of Virtual Sound Source positioning using HRIR coefficients. In *POSTER 2013 - 17th International Student Conference on Electrical Engineering*. Prague: Czech Technical University, 2013, p. 1-5. ISBN 978-80-01-05242-6.
- [7] *Home of the Chameleon*. [online]. URL:< <http://www.chameleon.synth.net>> Soundart. 2008.
- [8] *Listen HRTF database*. [online]. URL:<<http://recherche.ircam.fr/equipements/salles/listen/index.html>> IRCAM 2003
- [9] Lindner, T. Optimalizace měření a zpracování HRTF. In: *21th Annual Conference Proceedings Technical Computing Prague 2013*. Praha: 2013
- [10] Štorek, D. Virtual Sound Source Positioning by Differential Head Related Transfer Function In: *Proceedings of the AES 49th International Conference: Audio for Games*. New York: Audio Engineering Society, 2013, p. 1-6. ISBN 978-0-937803-90-5.

František Rund

Katedra radioelektroniky, FEL ČVUT v Praze, Technická 2, 166 27 Praha 6, Czech Republic
email: xrund@fel.cvut.cz

Martin Švejda

Katedra radioelektroniky, FEL ČVUT v Praze, Technická 2, 166 27 Praha 6, Czech Republic,