

# Lokalizace objektů v reálném čase

Štěpán Rydlo\*

## Abstrakt

Pro zjištění polohy objektu existuje řada metod a systémů. Většina těchto metod používá pro určení lokalizaci hardwarové implementace zpracování signálu. Zpracování signálu však lze řešit pomocí softwaru, který bude spuštěn na počítači. K počítači bude připojen rádiový přijímač. Cílem této práce je provést paralelizaci programu tak, aby zpracování informací probíhalo v reálném čase. Pro dosažení zpracování v reálném čase bude paralelizovat na CPU ale i GPU. Obsahem je popis dvou na sobě nezávislých možností přenosu dat a jeho zpracování.

**Klíčová slova:** lokalizace objektů — TDOA — OpenCL

## Přiložené materiály:

\*[xrydlo05@fit.vutbr.cz](mailto:xrydlo05@fit.vutbr.cz), Faculty of Information Technology, Brno University of Technology

## 1. Úvod

V dnešní době je používána lokalizace pro navigaci nebo různá měření. Pro zjištění polohy objektu však existuje řada metod a systémů. Rozvojem bezpilotních letadel – dronů pro využití např. střežení objektů, monitorování dopravní situace apod., vyvstává potřeba zjišťování jejich přesné polohy v reálném čase.

Současný rozvoj elektroniky a informačních technologií dává možnost vytvářet nové nezávislé systémy a zpracovávat rádiový signál pomocí počítačů. Vytvořené programy pro zpracování rádiového signálu se označují jako softwarová rádia a problém nastává při zpracování dat a určení polohy v reálném čase. Výhodou softwarového rádia je řešení právě pomocí programu. Všechny prvky pro zpracování signálu jsou řešeny softwarově. Proto s jedním přijímačem můžeme zpracovávat mnoho různých implementací bez zasahování do hardwaru. Výhodou softwarového řešení je cena, kdy používáme pro zpracování signálu počítač a nepotřebujeme tak různé hardwarové filtry nebo další prvky.

Existujícími řešeními jsou globální navigační družicové systémy (GNSS) [8]. Mezi tyto systémy patří GPS, GLONASS a vyvíjený systém GALILEO. Další možností pro lokalizaci jsou pozemní lokalizační systémy (LORAN). Výhodou systémů GNSS od pozemních lokalizačních systémů je dostupnost. Systémy GNSS se používají pro navigaci v automobilech, lodích nebo

i letadlech. Nevýhodou však je závislost na provozovateli těchto zařízení. Řešením této nevýhody je právě ve vytvoření vlastního lokalizačního systému zcela nezávislého na ostatních.

Výpočet polohy provádí program, který zpracovává data z rádiového přijímače připojeného k počítači. Lokalizační data přenášena rádiovými vlnami jsou vysílána několika vysílači. Každý z vysílačů přenáší jiný proud dat neboli sekvenci. Na základě zpoždění jednotlivých sekvencí na přijímači, je následně vypočítána poloha přijímače. Pro správné určení této polohy je zapotřebí znát polohy vysílačů, frekvenci a vysílané sekvence.

Pro dosažení zpracování dat v reálném čase na počítačích bylo využito grafických karet. Grafické procesory dosahují vyššího výpočetního výkonu než procesory CPU. Vytvořili jsme dvě na sobě nezávislé možnosti implementace lokalizačních systémů. Oba tyto systémy používají stejný způsob přenosu mezi vysílačem a přijímačem, rozdíl mezi těmito systémy je však v jejich implementaci přijímače a přesnosti. Oba tyto systémy jsou však nezávislé na existujících systémech.

Zpracování rádiových vln pomocí programu je výhodnější z hlediska přizpůsobení se lokalizačnímu systému. Kdybychom zpracovávali rádiový signál hardwarovými prvky v případě změny signálu v lokalizačním systému, by musela být provedena hardwa-

rová změna přijímače. Tato možnost může být časově náročná a složitá. Používáme program, který stačí v případě úpravy systému změnit zdrojový kód. V případě více lokalizačních systémů, kdy každý systém posílá data jiným způsobem, můžeme tuto možnost implementovat pouze pomocí změny programu a nemusíme vytvářet nové hardwarové zpracování.

## 2. Princip používaných lokalizačních systémů

Technologie lokalizace, co se týče její přesnosti a dostupnosti, dosáhla velkého pokroku v první polovině minulého století, kdy se pro lokalizaci začalo využívat rádiových vln. Do té doby se určovala poloha a směr podle hvězd, uměle vytvořených orientačních bodů a magnetických kompasů. Pro určení polohy museli být tyto body viditelné. Tyto metody však dosahují nízké přesnosti.

První lokalizační systémy používající rádiové vlny pro přenos informací byly pozemní systémy. Tyto systémy využívají pro určení polohy několika vysílačů. Nevýhodou těchto systémů však byla dostupnost. Při použití pozemních vysílačů dochází k omezení dostupnosti vlivem okolního terénu. Tento problém byl vyřešen vytvořením družicových lokalizačních systémů, které pro určení polohy využívají družic obíhající kolem Země ve výšce kolem 20 000 km. Tyto družice obíhají v několika orbitách a jejich počet a rozložení je upraveno tak, aby bylo dosaženo co nejvyššího pokrytí [8].

Dříve družicové navigační systémy využívali pro výpočet polohy Dopplerovu metodu. Tato metoda určuje, která z družic se přibližuje a která vzdaluje vlivem Dopplerova jevu. Na základě těchto informací byla vypočítána poloha přijímače a směr posuvu přijímače. Pomocí této metody však byla možnost určit polohu s přesností stovek metrů. Pro zvýšení přesnosti se začala používat dálkoměrná metoda za použití kódového přenosu informací. Přenášené informace při použití této metody je opakující se sekvence pseudonáhodných znaků, kde každý z vysílačů vysílá jinou předem stanovenou sekvenci. Pseudonáhodné znaky v sekvenci tvoří hodnoty 1 a -1. Tato sekvence je přenášena pomocí rádiových vln a přenášené hodnoty mění fázi signálu. Přijímač následně porovnává přijímaný signál se stejnou sekvencí, kterou si generuje. Pomocí tohoto porovnávání zjistíme jak dlouho trvalo přenesení sekvence od vysílače k přijímači.

Pro určení polohy touto metodou potřebujeme tedy znát přesný čas a polohu družic. Družice posílají svoji dráhu oběhu pomocí jiné frekvence. Následně pomocí přesného času potom může být vypočítána poloha

těchto družic v daný čas. Dále přesný čas potřebujeme pro jistění časového rozdílu mezi vysláním a přijetím sekvence. Polohu tyto systémy definují podle zeměpisné šířky ( $x$ ), zeměpisné délky ( $y$ ) a nadmořské výšky ( $z$ ). Pokud nepotřebujeme znát výšku stačí pro zjištění polohy signál ze tří vysílačů, v případě výšky potřebujeme signál z dalšího vysílače. Pro každý z vysílačů je vytvořena rovnice 1, kde  $x_i, y_i, z_i$  jsou definované polohou vysílače a hodnota  $\tau_i$  je rozdíl času mezi odesláním a přijetím lokalizační sekvence. Při doplnění těchto hodnot ze čtyř vysílačů nám vznikne soustava čtyř rovnic o čtyřech neznámých.

$$\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = c\tau_i \quad (1)$$

Jelikož pro určení polohy je zapotřebí více vysílačů a všechny tyto vysílače používají pro přenos rádiové vlny, musí být definován způsob jak rozlišit signál z jednotlivých vysílačů. Existují tři metody multiplexu, které definují tento způsob rozlišení. Při použití metody FDMA musí každý z vysílačů vysílat na jiné frekvenci, tak nebude docházet k vzájemnému rušení. Druhou možností je CDMA, kde jsou data z vysílačů přenášena na stejné frekvenci, ale každý z vysílačů vysílá jiná data. Poslední možností je metoda TDMA, kde každý z vysílačů vysílá v jiný předem stanovený čas. Pro lokalizaci jsou používány metody multiplexu FDMA a CDMA.

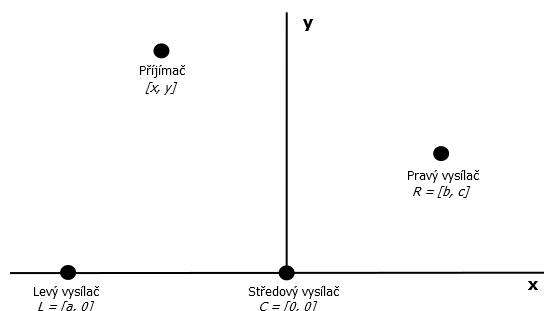
## 3. Možnost implementace lokalizačního systému

Pro vytvoření vlastního lokalizačního systému budeme vycházet z existujících systémů. Pro přenos dat mezi vysílačem a přijímačem použijeme rádiové vlny, kde k přenosu informací používáme modulaci BPSK. Tato modulace je založena na změně fáze přenášeného signálu. Tuto metodu používají systémy GNSS. Systém bude složen z několika pozemních vysílačů, kde každý z vysílačů bude přenášet jinou sekvenci na stejné frekvenci. Tyto pseudonáhodné sekvence jsou generovány pomocí Goldova generátoru. Stejný způsob generování sekvence používá systém GPS.

### 3.1 Výpočet polohy přijímače

Pro vypočítání polohy použijeme metodu TDOA [1], tato metoda se u systému GNSS nepoužívá. Při použití této metody musí všechny vysílače vysílat stanovenou sekvenci ve stejný čas. Podle rozdílů příchodu těchto signálů je následně určena poloha přijímače. Pro určení polohy přijímače potřebujeme znát pozice jednotlivých vysílačů. V našem případě se jedná o pozemní lokalizační systém, kde vysílače nemění

svoji pozici, proto tyto informace jsou statické a není potřeba je vypočítávat. Příklad jednoho takového systému je zobrazen na obrázku 1.



**Obrázek 1.** Příklad lokalizačního systému

Pro výpočet polohy použijeme Časoměrně-hyperbolickou metodu určování [2]. Výsledkem výpočtu této metody na základě informací z obrázku 1, budou dva průsečíky hyperbol. Tyto průsečíky určují polohu přijímače. Abychom zjistily, který z těchto průsečíků, je skutečnou polohou přijímače, musíme vypočítat vzdálenost mezi vypočítaným bodem a jednotlivými přijímači. Vzdálenost následně vydělíme rychlostí světla a výsledné rozdílné časy porovnáme s naměřenými hodnotami. Pokud znaménka hodnot naměřených a vypočítaných souhlasí jedná se o pozici přijímače.

### 3.2 Možnosti paralelizace implementace

Pro dosažení zpracování signálu a určení polohy přijímače v reálném čase musíme program vhodně paralelizovat. Polohu přijímače bude vypočítávat program, který bude spuštěn na počítači. Tyto zařízení mají procesory CPU, na kterých můžeme paralelizovat pomocí vláken nebo procesů.

V případě našeho programu je lepší provádět paralelizaci pomocí vláken. Vlákna na rozdíl od procesů dosahují rychlejší přenos dat, který bude zapotřebí. Pro synchronizaci vláken se používají mutexy, nebo podmínkové proměnné. V případě chyby v jednom vlákně může dojít k chybě v celém programu. U procesů k této chybě nemůže dojít, protože každý z procesů je kopií programu. Pro přenos dat mezi procesy musíme používat sdílenou paměť.

Dalším prvkem, který se nachází v počítačích, jsou grafické procesory GPU. Procesory CPU rychleji zpracovávají složité operace nebo skoky v instrukcích. CPU dosahují vyšších kmitočtů, mají větší mezipaměť, která umožňuje rychlé zpracování dat na jednom jádru [9]. Jádro v procesoru tvoří aritmeticko-logická jednotka. Aby program mohl vykonávat na více jádrech, musí být program paralelizován. Grafické karty však obsahují mnohem více aritmeticko-logických jednotek, které provádějí operace. Díky vyššímu počtu těchto

jednotek u grafických procesorů, dosahují grafické procesory vyšší výpočetní rychlosti než je tomu u procesorů CPU. Výpočetní rychlost se určuje v jednotkách FLOPS, tedy počet operací v plovoucí desetinné čárce za sekundu.

Způsob využití grafických karet k výpočtu obecných algoritmů se označuje GPGPU. První možnosti implementace obecných algoritmů na grafických procesorech přišly roku 2000, kdy standard Direct3D umožňoval programovatelné „shadery“. Aktuálními standardy pro programování grafických karet použitím GPGPU jsou CUDA pro grafické procesory nVidia, AMD Stream pro grafické procesory ATI a OpenCL. Standard OpenCL [3] umožňuje pracovat s grafickými procesory nVidia i ATI. Nicméně grafické procesory nVidia podporují OpenCL pouze po verzi 1.2, v dnešní době je však vydána verze 2.2.

## 4. Implementace lokalizace

Pro implementaci jsme zvolili programovací jazyk C. Pro paralelizaci na grafické kartě jsme použili standard OpenCL, který je založen na jazyku C. Objektově orientované programování pro tento projekt nebylo zapotřebí. Nastavení přijímače RTL-SDR, které je připojeno přes USB, probíhá pomocí funkcí, které poskytují již existující knihovny. Pro čtení dat z RTL-SDR musíme použít asynchronní nahrávání rádiového signálu. Při použití synchronního nahrávání dojde k nahrávání pouze ve chvíli, kdy je volána funkce pro čtení. Při použití synchronního čtení a ukládání dat do souboru docházelo ke ztrátě přibližně 70% dat.

Existuje řada přijímačů, které mohou být připojeny přes USB, ale i PCI. Implementaci je třeba vytvářet na předem určený přijímač. Posílaná data z přijímače a maximální vzorkovací frekvenci určuje digitální demodulátor, který obsahuje A/D převodník. Pro tento projekt jsme zvolili přijímač, který obsahuje demodulátor RTL2832U od firmy Realtek. Posílá I a Q vzorky signálu [4], které jsou zapsány v hodnotách beznámenkového celého 8-bitového čísla (`uint8_t`). Data jsou nahrávána do předem definovaného pole.

### 4.1 Implementace statického systému

Získání lokalizačního proudu dat ve statickém systému je tvořeno z několika na sobě závislých bloků. Prvním z bloků je přijímání signálu. Následuje prvek na demodulaci dat. Tento prvek zpracovává přijatý signál, jehož výstupem budou hodnoty 1 a -1 podle změny fáze. Pro toto zpracování používáme costasovu smyčku [5].

Tato získaná data jdou do korelátoru, který porovnává přijatá data s vygenerovanými referenčními sek-

vencemi. Tento prvek jsem implementovali na CPU, a výslednou rychlost zpracování bez paralelizace pro několik procesorů, můžeme vidět v tabulce 1. Na základě těchto výsledků je jasné, že zpracování v reálném čase bez paralelizace není možné. Z důvodu vyššího výpočetního výkonu byla paralelizace provedena na grafickém procesoru, jejíž výsledek je také možno vidět v tabulce 1.

CPU	Průměrná délka testu (s)
AMD Phenom II X4 840	71,146
Intel Core i7-4790	20,470
AMD FX-7500	48,964
Intel Core i7-6700HQ	28,706
GPU	
AMD Radeon HD 7660G	2,339
AMD Radeon HD 7600M	2,681
AMD Radeon R7 M260DX	0,652

**Tabulka 1.** Časové výsledky testů pro korelaci, 200 000 vzorků signálu. Pro reálný čas potřeba zpracovat za 0,1 s.

Poslední částí je nalezení vrcholů korelačních křivek pro jednotlivé vysílače. Na základě časových rozdílů mezi vrcholy jednotlivých přijímačů je určena poloha přijímače.

## 4.2 Implementace průběžného systému

Předchozí implementovaný systém pro statickou synchronizaci porovnává celou sekvenci a následně hledá výsledky. Tento systém neporovnává celou masku, ale pouze část. Pro porovnávání používá metodu, která se nazývá *early late correlation* [7]. Porovnává signál podle tří stejně vzdálených částí v referenční sekvenci. Na základě rozdílů průměru dřívějšího a pozdějšího porovnávání dojde k posunu referenční masky. Tato implementace tedy průběžně posouvá masku, dokud nedosáhne synchronizace s daty z přijatého signálu. Blokované schéma této implementace můžeme vidět na obrázku 2.

Tento způsob implementace se skládá z části *Code Tracking*, která porovnává a posouvá referenční masku. Druhou částí je *Carrier Tracking*, jehož úkolem je určit frekvenci amplitudy, ve které jsou přenášena data a zjistit zda-li je referenční sekvence synchronizována se signálem. Rozdíl posuvu mezi jednotlivými maskami určuje časový rozdíl pro vypočítání polohy přijímače. Pro každou sekvenci, kterou chceme porovnat v přijatém signálu, musíme vytvořit zvlášť tuto implementaci.

## 5. Závěr

Při provedení paralelizace statického systému na grafický procesor bylo dosaženo rychlejšího zpracování lokalizačních dat než tomu bylo při testech na CPU. Nicméně pro dosažení zpracování v reálném čase to nestačí, jak je zobrazeno v tabulce 1. Vyššího zrychlení můžeme ještě dosáhnout pomocí rychlejší grafické karty, nebo použitím více grafických karet. Pro určení polohy v reálném čase je zapotřebí signál z více než tří vysílačů. Nicméně při použití dvou nebo více vysílačů dochází ke ztrátě lokalizačních dat vlivem vzájemného rušení vysílačů. Proto tato implementace zůstala pouze pro simulační účely.

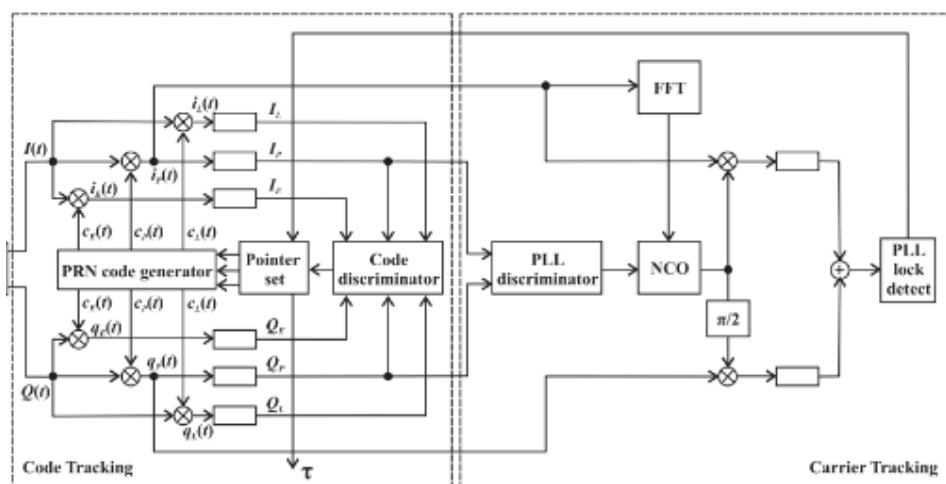
V případě signálu použitého pro systém s průběžnou synchronizací vzájemné rušení neovlivňuje data. Tento systém je méně hardwarově náročný. Jeho složitost ale stoupá s počtem porovnávaných sekvencí. Systém pro průběžnou synchronizaci se zatím vyvíjí, proto nejsou k dispozici žádné prokazatelné testy. Přesnost obou těchto systémů, co se týče vypočítané polohy, je výrazně ovlivněna vzorkovací frekvencí přijímače. Použitý přijímač dosahuje maximální vzorkovací frekvence 2MHz. Z toho můžeme určit, že přesnost určené pozice přijímače je 150 metrů. Zvýšit přesnost můžeme pomocí jiného přijímače, který dosahuje vyšší vzorkovací frekvence. Se zvýšením vzorkovací frekvence, se také zvyšují nároky na hardware, aby docházelo ke zpracování dat v reálném čase. Další možností pro přesnější určení polohy je multilaterace [6].

## Poděkování

Tímto bych chtěl poděkovat panu doc. Dr. Ing. Dušanu Kolářovi za umožnění a pomoci při zvolení postupu pro vypracování této práce. Dále bych chtěl poděkovat dalším lidem, kteří na tomto projektu se mnou pracovali, za vysvětlení a popis možností implementace.

## Literatura

- [1] *Základy teorie pasivních systémů I*. Univerzita obrany Brno, 2007.
- [2] *Základy teorie pasivních systémů I*, kapitola Časoměrně – hyperbolická metoda určování polohy zdroje signálu. Univerzita obrany Brno, 2007.
- [3] Introduction to OpenCL™ Programming. [online], 2010, Naposledy navštíveno 23. 1. 2017. URL <https://www.khronos.org/registry/OpenCL/specs/opencl-2.2.pdf>
- [4] What is I/Q Data? [online], 30. 3. 2016, Naposledy navštíveno 18. 1. 2017.



**Obrazek 2.** Blokové schéma implementace pro zpracování rádiového signálu

URL <http://www.ni.com/tutorial/4805/en/>

- [5] Hagemann, E.: The Costas Loop - An Introduction. [online], 2001, Naposledy navštíveno 18. 1. 2017.

URL <http://dsp-book.narod.ru/costas/DSP010315F1.pdf>

- [6] pplk. Ing. Petr Hubáček: *Optimalizace topologie TDOA systémů z hlediska přesnosti utčení polohy cíle*. Dizertační práce, Univerzita obrany v Brně, Fakulta vojenských technologií, 2010.

- [7] Navipedia: Delay Lock Loop (DLL). [online], 2014, Naposledy navštíveno 16. 3. 2017.

URL [http://www.navipedia.net/index.php/Delay\\_Lock\\_Loop\\_\(DLL\)](http://www.navipedia.net/index.php/Delay_Lock_Loop_(DLL))

- [8] Wikipedia: Satellite navigation — Wikipedia, The Free Encyclopedia. 2017, [Online; accessed 6-April-2017].

URL [https://en.wikipedia.org/w/index.php?title=Satellite\\_navigation&oldid=773867042](https://en.wikipedia.org/w/index.php?title=Satellite_navigation&oldid=773867042)

- [9] Zachar, M.: Co je to hardwarová akcelerace? [online], 27. 9. 2010, Naposledy navštíveno 18. 4. 2017.

URL <http://magazin.stahuj.centrum.cz/co-je-to-hardwarova-akcelerace/>