

UNIVERZITA PARDUBICE
DOPRAVNÍ FAKULTA JANA PERNERA

**METODY PRO VÝPOČET
VZÁJEMNÉ FUNKCE NEURČITOSTI**

DISERTAČNÍ PRÁCE

2012

Ing. Jan Pidanič

UNIVERZITY OF PARDUBICE
JAN PERNER TRANSPORT FACULTY

**METHODS FOR COMPUTING
CROSS AMBIGUITY FUNCTION**

DOCTORAL DISSERTATION

2012

Ing. Jan Pidanič

PROHLÁŠENÍ

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 15. března 2012

Jan Pidanič

ANOTACE

Disertační práce se zabývá analýzou algoritmů vzájemné funkce neurčitosti CAF, která hraje stěžejní roli pro určení odhadu bistatické vzdálenosti a radiální rychlosti v systémech pasivní koherentní lokace PCL. Náplní práce je popis, analýza a následná optimalizace výpočtu vybraných algoritmů CAF pro jednotlivé hardwarové konfigurace se zaměřením na rychlost výpočtu, což je stěžejní parametr pro vývoj radarových systémů na principu PCL. Každý algoritmus je podroben numerické simulaci a reálnému testování v jednotlivých hardwarových konfiguracích. Z provedených analýz výsledků vyplývají doporučení na volbu algoritmu CAF pro konkrétní HW konfiguraci.

KLÍČOVÁ SLOVA

pasivní koherentní lokace, bistatické radary, vzájemná funkce neurčitosti, cluster, GPU, Multi-core, číslicové zpracování signálů, paralelizace a vektorizace algoritmů

ANNOTATION

This doctoral dissertation deals with the analysis of algorithms for cross ambiguity function CAF, which plays a vital role in the estimation of bistatic range and radial velocity in passive coherent location PCL. This thesis aims at the analysis, description, and optimization of selected algorithms for individual hardware configuration with a focus on computation speed which is the primary parameter for the design and development of radar system based on this principle. Each algorithm is evaluated by means of numerical simulation and real time testing for various Hardware configurations followed by the recommendation for selection of specific algorithm of CAF for the particular hardware configuration.

KEYWORDS

Passive coherent location, bistatic radar, cross ambiguity function, cluster, GPU, Multi-core, digital signal processing, parallelization and vectorization of algorithms

PODĚKOVÁNÍ

Na tomto místě bych rád poděkoval všem, kteří mě podporovali během studia a následného psaní této disertační práce, bez jejichž podpory by tato práce nevznikla. Školiteli Ing. Jiřímu Konečnému, Ph.D., Prof. Ing. Pavlu Bezouškovi, CSc., Ing. Zdeňku Němcovi, Ph.D. a mnohým jiným za neocenitelné rady, řešení problémů a povzbuzení nutných pro úspěšné napsání této práce. Zvláštní poděkování pak patří mé rodině za její bezbřehou trpělivost a schovívavost, kterou projevovala zejména v době psaní tohoto textu.

OBSAH

SEZNAM ZKRATEK	9
1 ÚVOD.....	1
2 CÍLE DISERTAČNÍ PRÁCE A ZPŮSOB ŘEŠENÍ.....	3
3 BISTATICKÉ RADARY A PASIVNÍ KOHERENTNÍ LOKACE	4
3.1 METODY URČENÍ POLOHY SLEDOVANÝCH OBJEKTŮ.....	4
3.2 BISTATICKÉ RADARY	5
3.3 PASIVNÍ KOHERENTNÍ LOKACE (PASSIVE COHERENT LOCATION).....	9
3.3.1 ANTÉNNÍ POLE	12
3.3.2 PŘEDZPRACOVÁNÍ SIGNÁLŮ.....	13
3.3.3 ADAPTIVNÍ FILTRACE	13
3.3.4 CAF - VZÁJEMNÁ FUNKCE NEURČITOSTI	13
3.3.5 DETEKCE CÍLŮ BISTATICKÉHO RADARU A EXTRAKCE JEHO PARAMETRŮ	14
3.3.6 ASOCIACE DAT A SLEDOVÁNÍ CÍLŮ.....	14
3.4 NUMERICKÁ NÁROČNOST PCL SYSTÉMU	14
3.5 BUDOUCNOST PCL SYSTÉMŮ	15
4 TEORETICKÉ ZÁKLADY PRO VÝPOČET CAF	16
4.1 DISKRÉTNÍ FOURIEROVA TRANSFORMACE	16
4.2 ANALYTICKÝ SIGNÁL, KOMPLEXNÍ OBÁLKA A HILBERTOVA TRANSFORMACE	16
4.3 DEFINICE OPTIMÁLNÍHO A PŘIZPŮSOBENÉHO FILTRU	20
5 VZÁJEMNÁ FUNKCE NEURČITOSTI	23
5.1 ODVOZENÍ MODELŮ SIGNÁLŮ PRO VÝPOČET CAF	24
5.2 DEFINICE VZÁJEMNÉ FUNKCE NEURČITOSTI.....	27
5.3 VLASTNOSTI CAF.....	29
5.3.1 MAXIMUM CAF	29
5.3.2 KONSTANTNÍ OBJEM CAF	29
5.3.3 SYMETRIE CAF	29
5.3.4 ŘEZ V MAXIMU $ CAF $ PŘI KONSTANTNÍM KMITOČTU $f = f_D$	30
5.3.5 ŘEZ V MAXIMU $ CAF $ PŘI KONSTANTNÍM ZPOŽDĚNÍ $\tau = \tau_{D0}$	32
5.4 CAF V DISKRÉTNÍ OBLASTI.....	34
5.5 METODY VÝPOČTU DISKRÉTNÍHO TVARU CAF	36

5.5.1	SUMAČNÍ METODA (SM).....	37
5.5.2	KORELAČNÍ METODA (XC).....	38
5.5.3	FFT METODA (FFT).....	40
5.5.4	MATICOVÁ METODA VÝPOČTU (MM).....	42
5.5.5	METODA VÝPOČTU CAF POMOCÍ FFT S DECIMACÍ SIGNÁLOVÉHO SOUČINU (FFTD).....	44
5.5.6	METODA VÝPOČTU CAF VE FREKVENČNÍ OBLASTI (FMGFD).....	47
5.5.7	METODA VÝPOČTU CAF POMOCÍ ČÁSTEČNÝCH KORELACÍ (FMCW).....	50
6	VÝPOČET FUNKCE NEURČITOSTI V REALTIME APLIKACÍCH	54
6.1	VÝPOČET CAF ZALOŽENÝ NA DĚLENÍ VSTUPNÍCH DAT DO BLOKŮ	55
6.2	VÝPOČET CAF S VYUŽITÍM VÍCEJÁDROVÝCH CPU, VÍCEPROCESOROVÝCH PC, CLUSTERU.	55
6.3	VÝPOČET CAF S VYUŽITÍM GRAFICKÉ KARTY GPU.....	58
6.4	SOFTWAREVÉ A HW VYBAVENÍ PRO PARALELNÍ PROGRAMOVÁNÍ.....	59
6.4.1	SOFTWAREVÉ VYBAVENÍ.....	59
6.4.2	HARDWAROVÉ VYBAVENÍ.....	60
7	MĚŘENÍ A ANALÝZA NUMERICKÉ NÁROČNOSTI CAF.....	61
7.1	METODOLOGIE TESTOVÁNÍ	61
7.2	VSTUPNÍ PARAMETRY.....	65
7.3	MĚŘENÍ DOB VÝPOČTU U JEDNOTLIVÝCH METOD CAF.....	66
7.3.1	SUMAČNÍ METODA DLE DEFINICE (SMD).....	66
7.3.2	SUMAČNÍ METODA I.....	67
7.3.3	SUMAČNÍ METODA II.....	70
7.3.4	KORELAČNÍ METODA.....	72
7.3.5	FFT METODA.....	75
7.3.6	MATICOVÁ METODA VÝPOČTU CAF.....	77
7.3.7	FFT METODA S DECIMACÍ SIGNÁLOVÉHO SOUČINU.....	80
7.3.8	METODA VÝPOČTU CAF VE FREKVENČNÍ OBLASTI.....	82
7.3.9	METODA VÝPOČTU CAF POMOCÍ ČÁSTEČNÝCH KORELACÍ.....	84
7.4	VIZUÁLNÍ POROVNÁNÍ VYPOČTENÝCH POVRCHŮ CAF PRO JEDNOTLIVÉ METODY VÝPOČTU.	86
7.5	CELKOVÉ VYHODNOCENÍ VÝPOČTU CAF NAPŘÍČ VŠEMI METODAMI.....	87
7.5.1	VLASTNOSTI JEDNOTLIVÝCH ALGORITMŮ CAF.....	88
7.5.2	POROVNÁNÍ Z HLEDISKA RYCHLOSTI VÝPOČTU PRO SÉRIOVÉ ZPRACOVÁNÍ.....	89
7.5.3	POROVNÁNÍ Z HLEDISKA RYCHLOSTI VÝPOČTU PRO VŠECHNY HW KONFIGURACE.....	91
8	ZÁVĚR.....	94
8.1	ANALÝZA HARDWARU PRO VÝPOČET CAF.....	94
8.2	ANALÝZA ALGORITMŮ CAF.....	95

8.3	MOŽNOSTI DALŠÍ PRÁCE	96
	LITERATURA.....	97
	SEZNAM OBRÁZKŮ	103
	SEZNAM TABULEK	104
	SEZNAM PUBLIKACÍ.....	106

SEZNAM ZKRATEK

3D	3-Dimensional	3 rozměrný (3 dimensionální)
3G	3rd generation mobile telecommunications	3 generace mobilních telefonů
AM	Amplitude Modulation	Amplitudová modulace
AM DSB	Amplitude modulation Double Side Band	Oboustranná amplitudová modulace
AM SSB	Amplitude Modulation Single Side Band	Jednostranná amplitudová modulace
AOA	Angle of Arrival	Určení směru příchodu signálu
CAF	Cross Ambiguity Function	Vzájemná funkce neurčitosti
CDMA	Code Division Multiple Access	Kódový multiplex
CFAR	Constant false alarm rate	Konstantní úroveň falešného poplachu
CPU	Central processing unit	Procesor
DAB	Digital Audio Broadcasting	Digitální rozhlasová technologie
DFT	Discrete Fourier Transform	Diskrétní Fourierova transformace
DSP	Digital Signal Processor	Digitální signálový procesor
DVB-T	Digital Video Broadcasting – Terrestrial	Standard pozemního televizního digitálního vysílání
ERP	Effective Radiated Power	Efektivní vyzářený výkon
FDOA	Frequency Difference of Arrival	Rozdíl frekvencí přichozích signálů
FFT	Fast Fourier Transform	Rychlá Fourierova transformace
FLOPS	Floating-point Operations per Second	Počet operací v plovoucí řádové čárce za sekundu
FM	Frequency Modulation	Frekvenční modulace
FT	Fourier Transform	Fourierova transformace

GPU	Graphics processing unit	Grafický procesor
GSM	Global System for Mobile Communications	Globální Systém pro Mobilní komunikaci
HDD	Hard Disk Drive	Pevný disk
HPC	High-performance computing	
HW	Hardware	Fyzické technické vybavení počítače
I/O	Input/output	Vstup/výstup
IEEE	The Institute of Electrical and Electronics Engineers	Institut pro elektrotechnické a elektronické inženýrství
OFDM	Orthogonal Frequency Division Multiplexing	Modulace využívající kmitočtové dělení kanálu
PC	Personal Computer	Osobní počítač
PCL	Passive Coherent Location	Pasivní koherentní lokace
SNR	Signal to Noise Ratio	Odstup signálu od šumu
SW	Software	Programové vybavení
TDMA	Time division multiple access	Časový multiplex
TDOA	Time Difference of Arrival	Rozdíl časů příchodu signálů
TO	Transmitters of Opportunity	Příležitostné (nespolupracující) vysílače

1 ÚVOD

Z historického hlediska tvoří pasivní radarové systémy poměrně rozšířenou skupinu systémů, které mají v České Republice dlouhou tradici. Při pohledu do historie se jedná o systémy PRP1/Kopáč (1963), Ramona (1979), Tamara (1987), Věra (1995), které pracují na principu měření časových rozdílů v příchozech přijímaných signálů, tzv. TDOA princip. S prudkým rozvojem výpočetní techniky (z důvodu nárůstu výpočetního výkonu) se do popředí dostávají i radarové systémy založené na principu bistatických radarů, tzv. systémy **Pasivní Koherentní Lokace (PCL)**, kde jednu z numericky nejnáročnějších částí tvoří výpočet vzájemné funkce neurčitosti CAF. Výstupem ze vzájemné funkce neurčitosti je určení tzv. bistatické vzdálenosti a radiální rychlosti cíle(ů), jakožto nezbytných parametrů pro získání souřadnic a směru letu (a rychlosti) cíle(ů). Rychlost numerického výpočtu CAF je jedním z nejnáročnějších úkolů, které brání rychlejšímu rozvoji PCL systémů, a to z důvodu nutnosti provádět výpočet CAF v „reálném“ čase.

Předložená disertační práce se zabývá analýzou a vzájemným porovnáním jednotlivých metod výpočtu vzájemné funkce neurčitosti pro různé hardwarové konfigurace. Práce je řazena do sedmi na sebe navazujících kapitol, kde jsou na úvod práce popsány cíle a motivace disertační práce. Následuje kap. 3 s popisem PCL systémů a jejich vztahu k bistatickým radarovým systémům. Kap. 4 obsahuje základní teoretické předpoklady, jako jsou vybrané matematické definice a pojmy z oblasti číslicového zpracování signálů, potřebné pro sjednocení použité terminologie uvedené v disertační práci.

Stěžejní část práce tvoří kapitoly 5 až 7. Kapitola 5 je věnována odvození a uvedení základních vlastností spojitě vzájemné funkce neurčitosti, které předchází podrobná analýza přímého a odraženého signálu, jakožto vstupních signálů pro vlastní výpočet CAF. Následuje popis diskrétní verze CAF, která je stěžejní pro nasazení v reálných systémech PCL s uvedením všech významných parametrů. Mezi tyto parametry patří dosah radarového systému, rozlišení v bistatické vzdálenosti a Dopplerově posunu atd. Významnou část této kapitoly tvoří podrobná analýza osmi metod výpočtu CAF, kde součástí každého popisu je i zjednodušený popis algoritmu.

Kapitola 6 je věnována otázce výběru vhodného hardwarového vybavení pro numerický výpočet CAF, a to jak z hlediska rychlosti výpočtu CAF, tak z hlediska spolehlivosti, konfigurovatelnosti hardwarového vybavení a v neposlední řadě i finančních nákladů spjatých s pořízením tohoto hardwaru.

Nejrozsáhlejší kapitola 7 je věnována celkovému vyhodnocení algoritmů CAF jak z pohledu vlastností, omezení, výhod či jejich nevýhod, tak i z pohledu použitého

výpočetního hardwaru pro výpočet CAF. Z důvodu přehlednosti a porovnání vlivu algoritmů na jednotlivých HW konfiguracích je nejdříve uvedena podrobná analýza jednotlivých algoritmů pro všechny HW konfigurace, kde toto vyhodnocení je zpracováno formou tabulek s odpovídajícím komentářem. Následuje závěrečné vyhodnocení algoritmů z hlediska doby potřebné pro výpočet napříč všemi HW konfiguracemi. Součástí závěrečného hodnocení je i doporučení pro budoucí rozšíření práce.

2 CÍLE DISERTAČNÍ PRÁCE A ZPŮSOB ŘEŠENÍ

Radarové systémy založené na principu pasivní koherentní lokace (PCL) jsou považovány za velice perspektivní oblast budoucích pasivních radarových systémů. Jednou z překážek jejich reálného nasazení a rozšíření je numerická náročnost zpracování signálů, například výpočtu vzájemné funkce neurčitosti (CAF), která hraje klíčovou roli pro odhad polohy detekovaných cílů. Pro nasazení v reálných systémech je nutné provést numerický výpočet této funkce v co nejkratším možném čase, což vyžaduje využití modifikovaných metod výpočtu CAF oproti definičnímu vztahu CAF.

Hlavním cílem disertační práce je analýza stávajících metod pro výpočet vzájemné funkce neurčitosti s ohledem na možnosti výpočtu v rozdílných hardwarových (HW) konfiguracích, které jsou následující

- Pracovní PC stanice
- Multi-core systémy
- Cluster
- GPU computing

Jednotlivé HW konfigurace vyžadují odlišný přístup z hlediska programovacích technik.

Cíle disertační práce lze rozdělit do několika navazujících částí, kterými jsou:

- Teoretický popis a analýza vlastností stávajících metod pro určení vzájemné funkce neurčitosti a jejich použití v pasivní radiolokaci;
- Simulace metod výpočtu CAF;
- Ověření metod výpočtu CAF ve výpočetním systému MATLAB;
- Analýza HW konfigurací vhodných pro výpočet vzájemné funkce neurčitosti;
- Optimalizace a paralelizace algoritmů CAF;
- Analýza metod výpočtu CAF pro jednotlivé HW konfigurace.

3 BISTATICKÉ RADARY A PASIVNÍ KOHERENTNÍ LOKACE

Radarové systémy (RS) jsou komplexní elektronická zařízení sloužící k zaměření a identifikaci objektů (cílů) pomocí elektromagnetických vln. Slovo RADAR vzniklo z anglického spojení slov **R**adio **D**etecting **A**nd **R**anging. Radarové systémy lze rozdělit do mnoha skupin, a to podle principu detekce cílů, podle oblasti nasazení daného radarového systému atd.

V této kapitole je uveden popis pasivních radarových systémů, které tvoří podskupinu tzv. bistatických radarových systémů. Z důvodu seznámení s problematikou pasivních radarových systémů je tato kapitola členěna do několika na sebe navazujících podkapitol. Na začátku jsou vysvětleny principy detekce cílů u RS následované základním popisem radarů pracujících na bistatickém principu (bistatické radary), na jejichž principu jsou založeny systémy pasivní koherentní lokace (PCL). Seznámení s bistatickými RS je zde uvedeno pro jejich významnou odlišnost oproti „běžným“ monostatickým radarovým systémům. Pro tyto RS je uvedeno geometrické rozložení typického systému s uvedením základních charakteristik, jejichž vysvětlení je nutné pro navazující kapitoly.

Větší část této kapitoly je věnována podrobnému seznámení s pasivní koherentní lokací, která obsahuje geometrické rozložení typického systému, vysvětlení principu detekce a blokové schéma PCL systému.

3.1 Metody určení polohy sledovaných objektů

Pasivní radarové systémy určují polohu sledovaných objektů pomocí několika základních metod, případně jejich kombinací.

První metoda se nazývá směroměrná AOA (**A**ngle **o**f **A**rrival) a pro určení polohy objektu využívá azimut a elevaci radarem detekovaného signálu vyzářeného nebo odraženého od objektu. Pro přesné určení 3D polohy objektů je nutné použití alespoň tří přijímačů nacházejících se na různých místech. Nevýhodou této metody je nutnost velkých směrových antén, které jsou finančně velmi nákladné a pro mobilní zařízení i nepraktické. Další metodou je metoda časoměrná TDOA (**T**ime **D**ifference **o**f **A**rrival), která pracuje na principu měření časových rozdílů ve zpoždění signálů přijímaných jednotlivými přijímacími stanicemi, kde signál je odražený či vyslaný sledovaným objektem. Z rozdílů naměřených časů a ze známé polohy jednotlivých stanic můžeme eliminovat poslední neznámou, kterou je čas vyslání signálu od cíle, a určit tak polohu cíle. Minimální počet stanic pro určení polohy objektu v 3D prostoru jsou čtyři přijímače. V reálných systémech bývá počet stanic vyšší, než jsou minimální požadavky, z důvodu větší přesnosti a zejména ochraně proti výpadku

jednotlivých stanic, kdy by porucha jedné přijímací stanice mohla vést k chybnému nebo nemožnému určení polohy. Jednotlivé stanice jsou od sebe vzdáleny od několika km až po desítky km. V porovnání se směroměrnou metodou je časoměrná metoda přesnější především v úhlovém rozlišení, a to zejména díky velké vzdálenosti přijímacích antén (při optimálním rozložení). Další její nespornou výhodou je absence velkých směroměrných antén. Výsledný časoměrný systém je mobilnější a levnější na výrobu, než systém směroměrný. Za jeho nevýhodu se dá považovat složitost synchronizace jednotlivých přijímačů, nepřesnost určení vzdálenosti cíle v případě, že cíl leží vně sítě stanic, a také výrazně menší přesnost v určení výšky (elevation). Obecně lze říci, že jednotlivé výhody systémů AOA a TDOA se vzájemně doplňují a nevýhody se ruší. Ideální je využití obou metod současně v jednom systému.

Poslední metoda použitá pro pasivní radary využívá Dopplerova jevu FDOA (**F**requency **D**ifference **o**f **A**rrival). Dopplerův jev představuje změnu frekvence a vlnové délky přijímaného signálu oproti vysílanému signálu způsobenou nenulovou vzájemnou rychlostí vysílače a přijímače. Jestliže pohyblivý zdroj vysílá signál s frekvencí f_0 , pak na přijímací stanici (uvažujeme stacionární) bude frekvence dána vztahem

$$f = f_0 \left(1 - \frac{v_r}{c} \right), \quad (3.1)$$

kde c ... je rychlost šíření elektromagnetických vln v dané látce (vzduch),

v_r ... relativní rychlost zdroje vůči přijímací stanici, kde $v_r < 0$ pro přibližující se zdroj.

Podmínkou pro funkci radaru na tomto principu je, aby stanice a sledovaný objekt měly mezi sebou nenulovou relativní rychlost. Přesnost a pohotovost Dopplerovských systémů je nižší než u časoměrné metody, a to díky nutnosti sledovat delší úsek tohoto pohybu. FDOA systémy se používají zejména jako doplněk k TDOA systémům, jako přídatný zdroj informace, a to z důvodu, že samotné FDOA systémy neumí určit jednoduše polohu cíle a také z důvodu zpřesnění rychlosti cíle.

3.2 Bistatické radary

První vyvíjené radarové systémy pracovaly na bistatickém principu [1], tj. s oddělenou vysílací a přijímací částí. Teprve s příchodem duplexeru, který umožňoval oddělení vysílací a přijímací části, se začaly rozvíjet monostatické radary – radary se společnou anténou, která je využívána pro vysílání i příjem signálu. V raných dobách měl monostatický radarový systém výhody proti bistatickému zejména v nižší komplexnosti systému, snadnější synchronizaci vysílací a přijímací části, celý radarový systém se nachází na jednom stanovišti.

Nicméně bistatické radary mají řadu výhod, které je s rozvojem techniky, pokročilými metodami číslicového zpracování signálů, vzrůstající rychlostí DSP procesorů a možností, využití GPS satelitů pro synchronizaci či rozvojem fázových anténních soustav opět „dostaly“ do popředí zájmu zejména v oblasti vojenských radarových systémů. Největšími výhodami v porovnání s monostatickými radary jsou

(1) Přijímač je pasivní, tj. nelze jej detekovat pomocí jiných elektronických zařízení

(2) Bistatické radary umožňují detekovat cíle využívající *stealth* [32], [33], [34] technologii, protože tato technologie je zejména určena jako obrana proti monostatickým radarům.

(3) Bistatické radary umožňují pokrytí rozsáhlé oblasti s nižšími finančními náklady než u monostatických radarů, a to při pokrytí stejné oblasti.

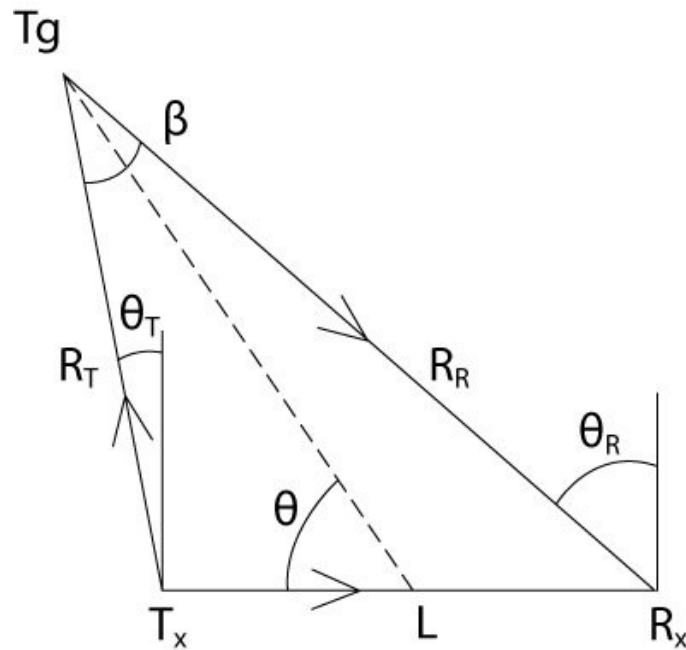
Cenu, kterou je nutné zaplatit za výhody bistatických systémů, je zejména nutnost přenosu velkého objemu dat do centrální stanice v případě koherentního zpracování signálu, vyšší komplexnost systému a obtížná synchronizace mezi jednotlivými částmi bistatického radarového systému. Pro větší názornost bude v následující části práce popsán princip bistatického radaru s popisem parametrů, které jsou důležité pro pasivní koherentní lokaci.

Bistatický radar je definovaný jako radar s oddělenou přijímací a vysílací anténou. IEEE definice bistatického radaru přesně nespecifikuje, jak daleko od sebe musí být přijímací a vysílací část [3]. Skolnik [4] toto oddělení definuje jako „značně velkou vzdálenost“. Oproti tomu Blake [2] definuje oddělení antén jako

(1) „směry přijímaného a vysílaného signálu od cíle se liší úhlem dopadu, který je srovnatelný nebo větší než šířka svazku antény“.

(2) „vzdálenosti cíle od vysílače a přijímače jsou různé a liší se významnou částí“. Jak je vidět z výše uvedených definic, definice bistatického radaru není jednoznačná a liší se u jednotlivých autorů odborné literatury.

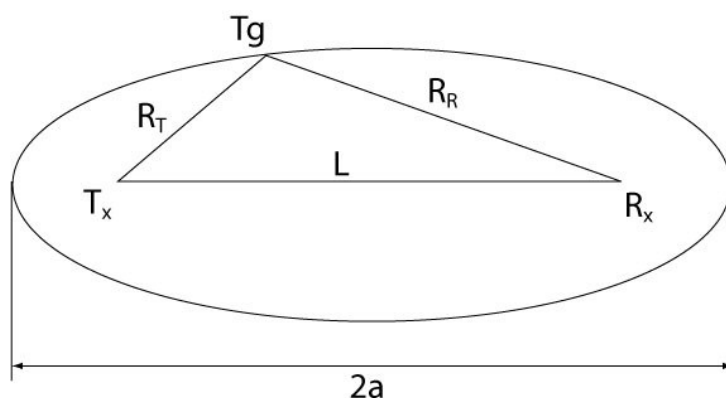
Základní geometrické rozložení bistatického radaru je trojice vysílač – cíl – přijímač, které tvoří tzv. bistatický trojúhelník. Na Obr. 1 můžeme vidět umístění přijímací stanice (R_x), vysílací stanice (T_x) a cíle (T_g). Přijímací i vysílací stanoviště může být na zemi, ve vzduchu či ve vesmíru a může či nemusí se pohybovat vůči Zemi. Vzdálenost $R_T = T_x - T_g$ je vzdálenost od vysílače k cíli, který je „ozářen“ vysílaným signálem, a cesta $R_T = T_g - R_x$ je cesta odraženého signálu k přijímači, kde je signál dále analyzován. Vzdálenost $L = T_x - R_x$ je tzv. základna a jedná se o vzdálenost mezi přijímací a vysílací částí. Úhel β je bistatický úhel.



Obr. 1 Bistatický trojúhelník

Detekce cíle(ů) je u bistatického radaru obdobná jako v případě monostatickém [4], [5], [6], [7]. Cíl T_g je ozářen vysílačem T_x a odražené signály od cíle jsou zachycovány přijímačem R_x , který je dále zpracovává. Vzdálenost vysílač – cíl – přijímač je tzv. bistatická vzdálenost $R_B = R_T + R_R$, a její určení zároveň s určením Dopplerova posunu je stěžejní úlohou u pasivních koherentních systémů, viz kapitola 2.

Bistatická vzdálenost určuje rotační elipsoid s ohnisky v místech vysílače a přijímače, na jehož povrchu se nachází potencionální cíl. Pro určení konkrétní polohy cíle je nutné mít k dispozici více dvojic přijímač-vysílač a z průniku povrchů jejich rotačních elipsoidů polohu cíle vypočítat. Úloha určení bistatické vzdálenosti a Dopplerovy rychlosti se pro jednotlivé dvojice přijímač – vysílač řeší nezávisle a proto zde budu dále uvažovat pouze jednu dvojici přijímač-vysílač. Elipsoid je definován pomocí dvou parametrů, a to vzdáleností mezi přijímačem a vysílačem L a parametrem a , kde vzdálenost $2a = (R_T + R_R)$ tvoří konstantní parametr součtu vzdáleností R_T a R_R (jiná označení jsou *isorange*). Provedeme-li řez v rovině bistatického trojúhelníku dostaneme elipsu, na které se může vyskytovat cíl.



Obr. 2 Elipsoid možných výskytů cílů

Na obrázku Obr. 2 je vykreslena pouze jedna elipsa, ale je zřejmé, že cíl nacházející se v jiné vzdálenosti od radaru bude produkovat jinou elipsu. Neplatí ovšem, že radar je schopný detekovat všechny cíle splňující podmínku $R_T + R_R = 2a$ v libovolné vzdálenosti od radaru, ale záleží na jeho maximálním dosahu.

Obecně schopnost radaru rozlišit a detekovat cíle v odražených signálech od různých objektů s určením jejich vlastností jako je velikost, vzdálenost, směr příchodu odraženého signálu převážně závisí na tvaru vysílané vlny a vlastnostech *clutteru*, který je definován jako nechtěné odrazy vyskytující se v signálu - odrazy od země, moře, deště, budov a jiných objektů [8].

Základní charakteristiky bistatického radaru, jsou obdobné charakteristikám monostatického radaru. Jedná se zejména o maximální dosah radaru, rozlišení radaru, které lze charakterizovat jako minimální „vzdálenost“, ve které lze dva či více cílů od sebe rozeznat v jedné či více charakteristikách (v souřadnicích x, y a z, úhlu, vzdálenosti, velikosti Dopplerova posunu způsobeného pohybem cíle (určení rychlosti cíle), zrychlení).

Maximální dosah radaru κ , je určen bistatickou radarovou rovnicí [1]

$$\kappa = (R_T R_R) = \left[\frac{P_T G_T G_R \lambda^2 \sigma_B F_T^2 F_R^2}{(4\pi)^3 k T_s B_n SNR_{\min} L_T L_R} \right]^{1/2}, \quad (3.2)$$

kde R_T je vzdálenost vysílač – cíl,

R_R je vzdálenost přijímač – cíl,

P_T je vysílaný výkon,

G_T je zisk vysílací antény,

G_R je zisk přijímací antény,

λ je vlnová délka,

σ_B je bistatická radarová odrazná plocha RCS (Radar Cross Section),

F_T je faktor šíření pro vzdálenost vysílač – cíl,

F_R je faktor šíření pro vzdálenost cíl – přijímač,

k je Boltzmannova konstanta,

T_s je šumová teplota systému,

B_n je šumová šířka pásma přijímacího filtru, přizpůsobeného přijímanému signálu,

SNR_{\min} je minimální odstup signálu od šumu pro detekci cílů,

L_T jsou ztráty vysílače (>1), které nejsou zahrnuty v jiných parametrech,

L_R jsou ztráty přijímače (>1), které nejsou zahrnuty v jiných parametrech,

κ je dosah radaru.

Bistatická radarová rovnice je obdobou radarové rovnice u monostatického radaru [4], [5] a vyjadřuje vzdálenost, ve které je konkrétní bistatický radar schopný detekovat určitý cíl. Podrobný popis jednotlivých proměnných je uveden v [3].

Podrobné informace věnující se bistatickým radarům lze najít v [1], [2], [3], [9]. Je nutno upozornit, že výše uvedené publikace se věnují tzv. aktivním bistatickým radarům, které se v mnoha ohledech od pasivních koherentních systémů odlišují. Využití popisu aktivních bistatických radarů spočívá zejména ve shodném rozložení (geometrii) aktivních i pasivních radarů.

3.3 Pasivní koherentní lokace (Passive Coherent Location)

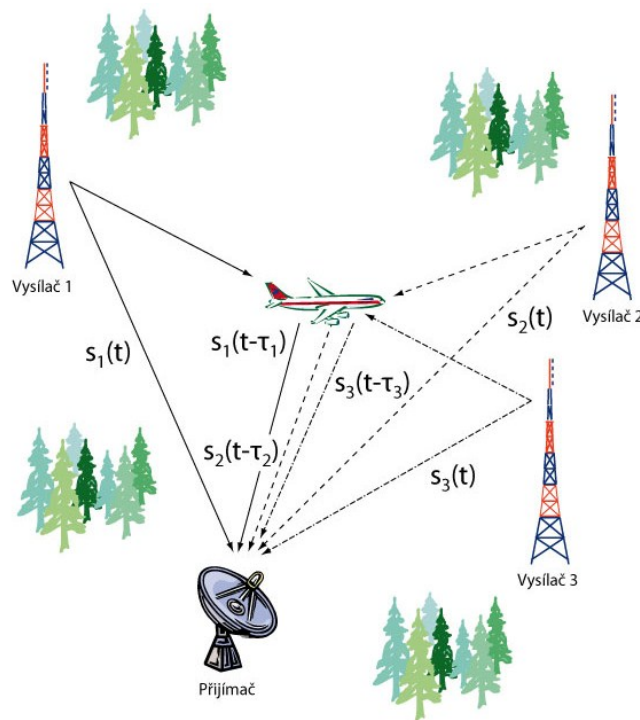
Pasivní radarové systémy využívají k detekci cílů elektromagnetické vlny, které se již vyskytují v okolním prostředí. Součástí pasivních radarových systémů tedy není vysílač, tyto systémy využívají (parazitují) na vysílačích, které jsou primárně určeny ke zcela jiným účelům. Díky absenci vysílače mají pasivní radary mnoho výhod, mezi něž především patří menší rozměry, vyšší mobilita, nižší cena, absence jakéhokoliv druhu stínění a pro funkčnost radaru není potřeba mít oprávnění pro provoz v určitém frekvenčním pásmu. Pasivní radary neemitují žádný signál a díky tomu je není možné detekovat.

Myšlenka využití vysílačů již nacházejících se v prostoru (tzv. nespolupracujících - příležitostných vysílačů – Transmitters of Opportunity - TO) není nová [13], [14].

Takovýto radarový systém je využíván a optimalizován pro detekci a sledování cílů pomocí televizních, rádiových, GSM a jiných vysílačů.

Typické rozmístění PCL systému je zobrazeno na Obr. 3, kde $s_x(t)$ a $s_x(t-\tau_x)$ jsou označení přímých a odražených signálů pro $x=1,2,3\dots N$, kde N je počet dvojic přijímač – vysílač.

Princip detekce cílů je založen na zpracování přímých (referenčních) signálů $s_x(t)$ a odražených signálů, které se šíří po spojnici vysílač – cíl – přijímač.



Obr. 3 Ukázka rozmístění PCL systému

Tento systém se tedy skládá z několika bistatických radarů, tvořených vždy přijímačem a jedním z vysílačů.

V tomto systému bude vysílaný signál náhodný proces daný druhem vysílače a právě vysílaným programem (např. řečí, hudbou, videem). Vysílaný signál u PCL systémů nelze ovlivnit a je nutné pracovat se signálem, který je k dispozici, což vede k větším nárokům na zpracování signálů oproti aktivním radarům, kde má konstruktér plnou kontrolu nad vysílaným signálem. Pro PCL systémy jsou využívány následující typy vysílačů:

- FM vysílače (ideální zdroj jak z hlediska pokrytí, tak i vyzářovaného výkonu) [15], [16], [17]
- DAB vysílače (malé pokrytí, teprve v začátcích) [18], [19], [20]

- Vysílače analogové televize (došlo k zastavení provozu z důvodu přechodu na digitální vysílání) [21], [22], [23]
- Vysílače digitální televize DVB-T [19], [24], [25], [26]
- GSM vysílače (malý vyzařovaný výkon směřovaný do dolní hemisféry prostoru, špatné pokrytí pro letící cíle) [27], [28], [29], [30]

Vybrané parametry, rozhodující o vlastnostech pasivního systému z hlediska výběru příležitostného vysílače, jsou velikost efektivního vyzařovaného výkonu vysílače (ERP), šířka pásma, druh použité modulace, atd. Srovnání jednotlivých vysílačů s nejdůležitějšími parametry jsou uvedeny v Tab. 1. Parametr ERP se liší pro každý vysílač, v tabulce jsou uvedeny hodnoty nejvyššího výkonu vysílače v ČR.

Druh vysílače	Kmitočet	Šířka pásma	Druh modulace	ERP
AM rozhlasové vysílače	639 kHz	9 kHz	DSB AM	1,5 MW
FM rozhlasové vysílače	95 MHz	150 kHz	FM	100 kW
Analogové Tv vysílače	~ 550 MHz	8 MHz	VSB AM (obraz) FM (zvuk)	*
DVB-T	~ 750 MHz	7,61 MHz	OFDM	100 kW
DAB	100 ~ 1500 MHz	**		
Mobilní síť GSM	900 MHz, 1.8 GHz	200 kHz	GMSK	80 W
Mobilní síť 3G	2 GHz	5 MHz	CDMA	80 W

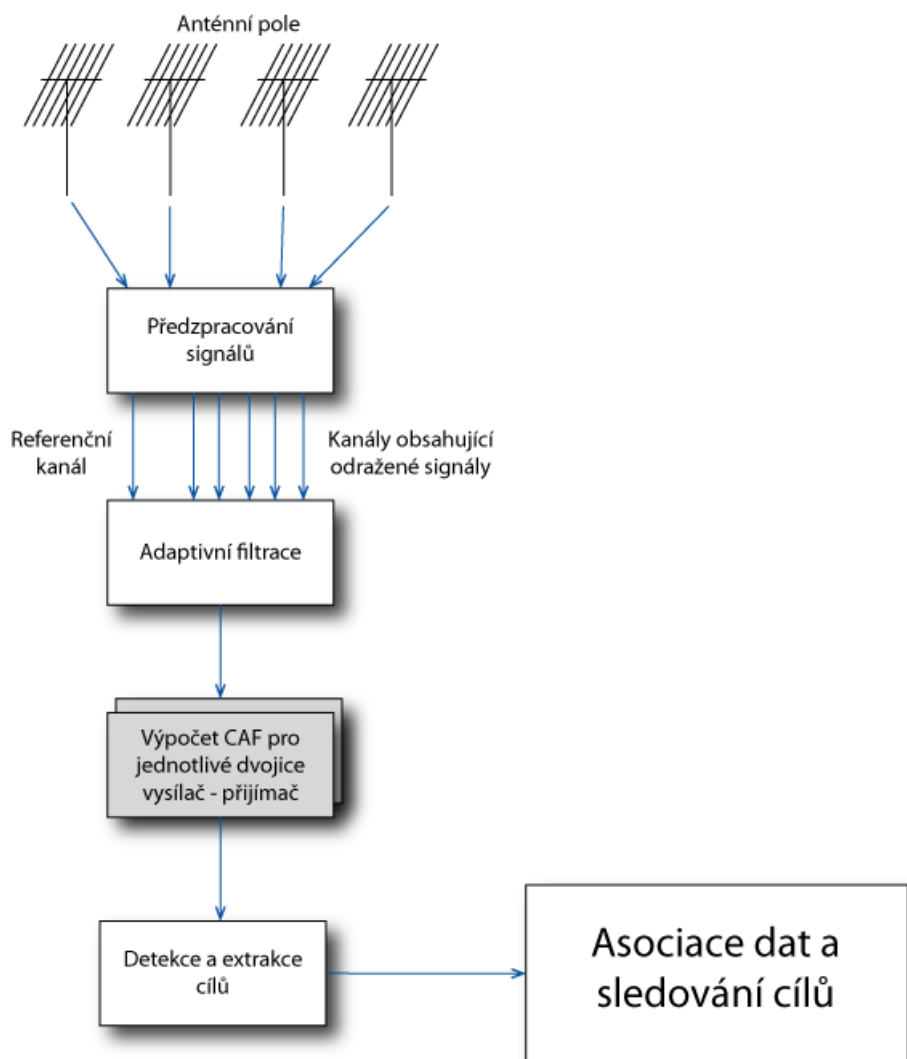
* vysílání v ČR ukončeno, ** standardy DAB jsou ve vývoji

Tab. 1. Seznam vysílačů pro různé služby v ČR [31]

Pasivní radarové systémy musí detekovat velice slabé odrazy od cílů ve velice silném rušení, což vede k následujícím požadavkům:

- nízké šumové číslo systému
- velký dynamický rozsah
- linearita systému

Blokové schéma přijímače pasivního radaru je zobrazeno na Obr. 4 [12]. V následujících odstavcích budou popsány jednotlivé bloky.



Obr. 4 Blokové schéma přijímače pasivního radaru

3.3.1 Antennní pole

Antennní pole slouží ke „sběru signálů“, které obsahují informace o cílech. Přijímací anténa je tvořena zpravidla více anténami dohromady, aby příslušný radarový systém pokryl zájmovou oblast (počet prvků v anténním poli bude záviset na typu prostoru, který chceme pokrýt). Každý prvek anténního pole tak pokrývá určitou část v prostoru.

3.3.2 Předzpracování signálů

Předzpracování signálů zahrnuje například analogovou filtraci signálu, oddělení signálů jednotlivých vysílačů, jejich digitalizaci, konverzi na komplexní obálky a další.

3.3.3 Adaptivní filtrace

Pro detekci cílů v pasivních radarových systémech je nutné mít k dispozici oddělený přímý (referenční) a odražený signál. Signál přijatý anténou ale obsahuje jak přímý signál, tak odražený signál a clutter. Adaptivní filtrace je použita k potlačení přímého (a daleko silnějšího) signálu a clutteru (tvoří tzv. statické odrazy na různých vzdálenostech) a tím k získání pouze odraženého signálu. Tento krok je důležitý z hlediska snížení možnosti překrytí hledaného cíle postranními laloky silnějšího signálu. Eliminace pozemního clutteru je extrémně důležitá. V případě nevhodně „nastavené“ filtrace lze ztratit informace o cíli a tím dojde k nemožnosti jej detekovat. Adaptivní filtrací se zabývají např. publikace [10], [11].

3.3.4 CAF - Vzájemná funkce neurčitosti

Výpočet vzájemné funkce neurčitosti lze považovat za jádro přijímače bistatického radaru pasivního radarového systému, které slouží k odhadu bistatické vzdálenosti a Dopplerových rychlostí cílů. Vzájemnou funkci neurčitosti lze často nalézt pod zkratkou CAF (z anglického **C**ross **A**mbiguity **F**unction) a v našem případě vyjadřuje korelaci mezi přímým (referenčním) signálem, který se šíří po spojnici vysílač-přijímač a odraženým (přijímaným) signálem šířícím se po spojnici vysílač-cíl-přijímač. Obecná definice CAF je dána vztahem:

$$CAF(\tau, \omega) = \int_{-\infty}^{+\infty} s_T^*(t) s_R(t + \tau_D) e^{j\omega t} dt, \quad (3.3)$$

kde $s_T(t)$ je vysílaný signál,

$s_R(t)$ je přijatý signál,

τ_D je časový posun signálů.

Tato funkce se pro každý bistatický radar systému vyhodnocuje zvlášť. Maxima $CAF(\tau, \omega)$ pro jeden bistatický radar leží v místech zpoždění τ_{Di} a Dopplerových posunů ω_{Di} jednotlivých cílů v dosahu radaru.

Výpočet této funkce je, jak bude ukázáno v dalších kapitolách, extrémně náročný z hlediska požadavků na výpočetní výkon. Hlavním cílem této disertační práce je vyhodnocení numerické efektivity výpočtu vzájemné funkce neurčitosti, analýza a optimalizace jednotlivých metod výpočtu CAF, která je stěžejní pro praktické

nasazení ve skutečných systémech pracujících v reálném čase. Podrobný teoretický rozbor CAF lze nalézt v kapitole 5.

3.3.5 Detekce cílů bistatického radaru a extrakce jeho parametrů

Na výstupu tohoto obvodu u každého bistatického radaru systému jsou odhadnuté parametry (tj. bistatické vzdálenosti a Dopplerovy rychlosti) všech relevantních cílů, zachycených tímto radarem. Při detekci se obvykle využívá metody CFAR určující úroveň, při které bude udržena požadovaná pravděpodobnost výskytu falešného poplachu. Zjednodušeně si lze tuto úroveň představit jako konkrétní velikost např. amplitudy či výkonu signálu, při jejímž překročení vydá detektor informaci o výskytu potenciálního cíle. Stanovení této úrovně vychází ze statistických parametrů vstupních signálů. Podrobné informace o určení této úrovně lze nalézt v [4], [6].

Extrakcí se v našem případě rozumí odhad souřadnic (τ_{Di}, ω_{Di}) těch maxim CAF, která byla detektorem propuštěna do dalšího zpracování.

3.3.6 Asociace dat a sledování cílů

Jak bylo uvedeno dříve, výstupem z jedné dvojice přijímač – vysílač je určení bistatické vzdálenosti a Dopplerovy rychlosti (R_b, v_r) . Pro určení souřadnic cíle ve 3D prostoru je nutné mít k dispozici více dvojic přijímač – vysílač. Další zpracování tedy probíhá společně pro výstupy všech bistatických radarů. Nejprve probíhá asociace dat jednotlivých bistatických radarů, tedy určení těch dvojic (R_b, v_r) od jednotlivých bistatických radarů, které odpovídají skutečným cílům. Přitom dochází k potlačení falešných cílů. K tomu se používá řada metod mj. i Kalmanova filtrace.

Dále probíhá sledování cílů, výpočet parametrů jejich drah, identifikace cílů a jejich zobrazení.

3.4 Numerická náročnost PCL systému

PCL systémy, založené na příjmu signálu nekooperujících vysílačů jako jsou rozhlasové nebo televizní vysílače jsou relativně nenáročné na vysokofrekvenční část HW: nejsou zapotřebí velké směrové antény, výkonné vysílače ani extrémně citlivé širokopásmové vysokofrekvenční obvody. Tato jednoduchost je vykoupena vyšší náročností na digitální část zpracování signálu, dat a informací. Od digitálního vícekanálového přijímače s vysokou dynamikou a ekvalizačních filtrů, oddělujících užitečný signál od přímého signálu a clutteru až po komplikovanou asociaci dat jednotlivých bistatických radarů v situaci mnoha sledovaných cílů. Mezi tyto složité bloky digitálního zpracování patří také blok výpočtu funkce neurčitosti (CAF).

Náročnost výpočtu CAF v tomto systému souvisí především s extrémní délkou úseků signálu, které je zapotřebí zpracovávat v časové a frekvenční oblasti a to v reálném čase. Výpočet CAF přitom bohužel nemá charakter klasické filtrace, ale vyžaduje blokové zpracování. Vysoká rozlišovací schopnost v Dopplerově kmitočtu zajišťující kvalitní separaci hledaných pohyblivých cílů a současně dostatečně vysoké potlačení šumu vyžadují zpracovávat dlouhé časové úseky signálu. Určení množství operací pro výpočet CAF je velice komplikované určit, a to z důvodu velkého rozdílu počtu operací mezi jednotlivými metodami výpočtu. Tyto požadavky vedou přirozeně k potřebě maximálního zefektivnění výpočetních postupů a k použití paralelizace výpočetních jednotek. A právě analýze, srovnání a ověření těchto metod výpočtu CAF je věnována tato práce.

3.5 Budoucnost PCL systémů

Vyššímu rozvoji PCL systémů brání zejména fakt, že řada potencionálních uživatelů (armáda, řízení letového provozu, apod.) považuje vysílače, na které nemá provozovatel PCL systému žádný vliv, za zcela nepřijatelnou variantu z důvodu spolehlivosti.

Budoucnost PCL systémů bude pravděpodobně ve využívání

(1) samostatných PCL systémů využívajících dedikovaných vysílačů, přizpůsobených pouze potřebám PCL systémů. Tyto vysílače budou vysílat přesně definovaný „pseudošumový“ signál, který nebude potřeba detekovat jako referenci, a to z důvodu, že tento signál bude znám (tento signál lze generovat na přijímači jako „kopii“ signálu vysílaného, či přenášet z vysílače pomocí datových linek k přijímači). Hlavní výhodou využití dedikovaných vysílačů je tedy eliminace detekce referenčního signálu a možnost přizpůsobení vysílaného signálu potřebám PCL zpracování. Nevýhodou je nutnost vývoje a provozu dedikovaných vysílačů.

(2) kombinace TDOA a PCL systémů s dedikovanými vysílači signálu. Využití kombinace těchto dvou systémů povede k univerzálnosti a schopnosti řešit širokou třídu uživatelských scénářů.

Obecnou snahou je získat z detekovaných signálů maximální množství informace a její následné využití při dalším zpracování dat.

4 TEORETICKÉ ZÁKLADY PRO VÝPOČET CAF

V této kapitole jsou uvedeny teoretické poznatky z oboru číslicového zpracování signálů a informací, které budou využívány v dalších kapitolách pro odvození jednotlivých algoritmů výpočtu CAF použitých v této práci. Jedná se především o využití diskrétní Fourierovy transformace, komplexní obálky signálu a přizpůsobeného filtru, jež jsou základními kameny pro odvození vzájemné funkce neurčitosti. Tato kapitola je zde uvedena také z důvodu unifikace pojmů používaných v této práci.

4.1 Diskrétní Fourierova transformace

Fourierova transformace (FT) slouží k transformaci signálu $s(t)$ v časové oblasti do oblasti frekvenční $S(\omega)$. Výstupem z FT je tzv. spektrum signálu. Definiční vztah pro přímou a zpětnou FT je dán vztahy

$$\begin{aligned} S(\omega) &= \int_{-\infty}^{\infty} s(t) \exp(-j\omega t) dt \\ s(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) \exp(j\omega t) d\omega. \\ S(\omega) &= F\{s(t)\} \end{aligned} \quad (4.1)$$

Pro numerické výpočty je spojitý tvar FT nevhodný, a proto zavádíme tzv. diskrétní (přímou a zpětnou) Fourierovu transformaci (DFT) definovanou vztahy

$$\begin{aligned} S_n &= \frac{1}{N} \sum_{k=0}^{N-1} s_k \exp\left(-j2\pi \frac{n}{N} k\right) \\ s_k &= \sum_{n=0}^{N-1} S_n \exp\left(j2\pi \frac{n}{N} k\right), \end{aligned} \quad (4.2)$$

kde S_n je posloupnost vzorků spektra signálu, N je celkový počet vzorků vstupního signálu, s_k je posloupnost vzorků vstupního signálu. V praxi se používá rychlá Fourierova transformace (FFT), která vychází z DFT, dochází ale k významné redukci numerických operací a tím i k významné časové úspoře při výpočtu. Podrobnější informace ohledně vlastností FT, DFT a FFT lze nalézt například v [35], [36], [37].

4.2 Analytický signál, komplexní obálka a Hilbertova transformace

Výsledkem Fourierovy transformace reálného pásmového signálu je komplexní spektrum signálu, které je komplexně sdružené podle osy y . Analytické vyjádření

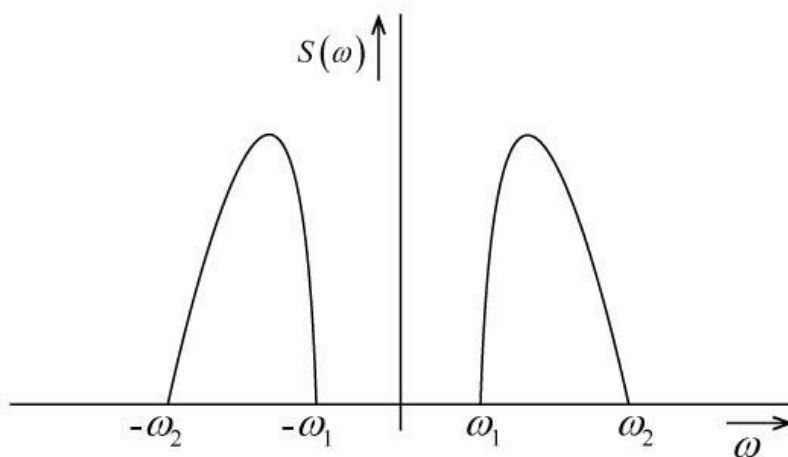
signálu vychází z premisy, že kompletní informace o signálu je obsažena jak v části spektra kladných frekvencí, tak i v části záporných frekvencí. Z tohoto důvodu je možné spektrální složky pro záporné frekvence odstranit bez ztráty informace o vstupním signálu [39].

Předpokládejme obecný vstupní radarový signál ve tvaru

$$s(t) = A(t) \cos[\omega_0 t + \varphi(t)], \quad (4.3)$$

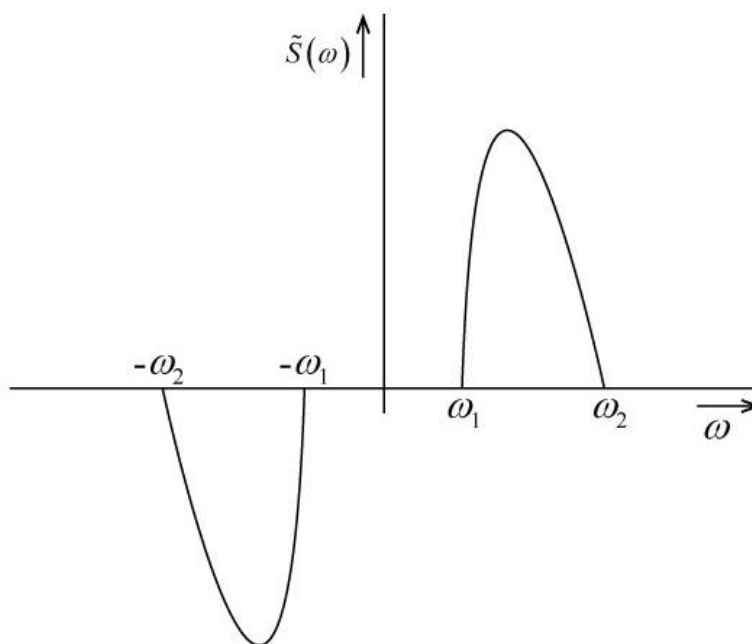
kde $A(t)$ je amplituda vstupního signálu závislá na čase, $\varphi(t)$ je fáze vstupního signálu závislá na čase, ω_0 je nosná frekvence signálu.

Pro názornost předpokládejme, že vstupní signál má spektrum, jež je zobrazeno na Obr. 5.



Obr. 5 Spektrum vstupního signálu

Z původního spektra signálu $S(\omega)$ vytvoříme pomocný signál $\tilde{S}(\omega) = S(\omega) \text{sign}(\omega)$, jehož spektrum je zobrazeno na Obr. 6.



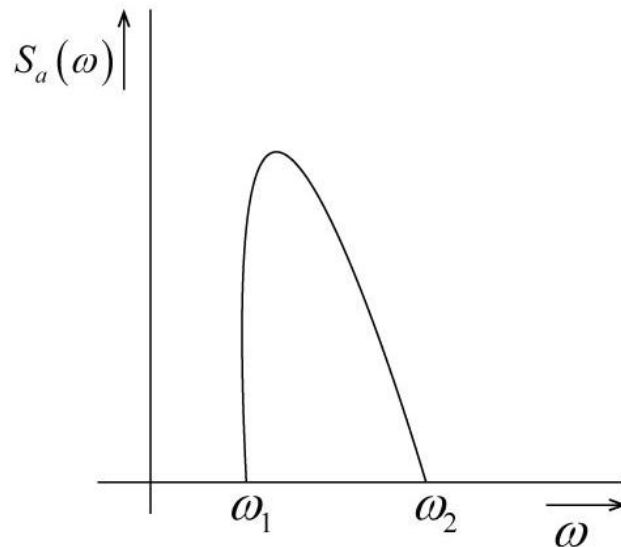
Obr. 6 Spektrum pomocného signálu $\tilde{S}(\omega)$

Tento signál již není reálný, ale čistě imaginární. Kdyby se jednalo o reálný signál, spektrum by muselo splnit podmínku $\tilde{S}(-\omega) = \tilde{S}^*(\omega)$ a ta splněna není. K získání reálného signálu vydělíme pomocný signál se spektrem $\tilde{S}(\omega)$ imaginární jednotkou j a vytvoříme tzv. sdružený signál $\hat{S}(\omega) = \frac{\tilde{S}(\omega)}{j}$, který již bude reálný.

Analytický signál pak bude definován jako součet signálu původního a signálu sdruženého

$$S_a(\omega) = S(\omega) + j\hat{S}(\omega) = S(\omega) + \tilde{S}(\omega) = \begin{cases} 2S(\omega), & f > 0 \\ 0, & f < 0 \end{cases} \quad (4.4)$$

Analytický signál lze definovat jako signál, jehož Fourierova transformace je pro záporné kmitočty nulová. Spektrum analytického signálu je zobrazeno na Obr. 7. Výhodou analytického vyjádření je, ve srovnání s reálným signálem, poloviční šířka pásma.



Obr. 7 Spektrum analytického signálu

Z analytického vyjádření signálu dostaneme původní reálný signál pouze vzetím reálné části analytického vyjádření

$$S(\omega) = \text{Re}[S_a(\omega)]. \quad (4.5)$$

Pro převod vstupního signálu na analytický tvar signálu v časové oblasti využijeme vlastnosti Fourierovy transformace, která nám říká, že násobení dvou signálů ve frekvenční oblasti je rovno konvoluci signálů v časové oblasti. S využitím této vlastnosti dostaneme pro výpočet sdruženého signálu vztah

$$\hat{s}(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{s(t')}{(t-t')} dt'. \quad (4.6)$$

Vztah (4.6) je definičním vztahem pro Hilbertovu transformaci, která nám převádí časový průběh signálu na signál sdružený. Analytický signál v časové oblasti je pak definován vztahem

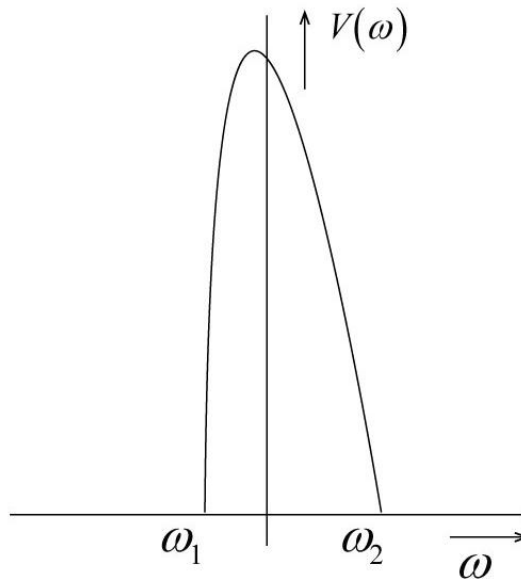
$$s_a(t) = s(t) + j\hat{s}(t) = s(t) + \frac{j}{\pi} \int_{-\infty}^{\infty} \frac{s(t')}{(t-t')} dt'. \quad (4.7)$$

Komplexní obálka signálu je definována pomocí analytického signálu, a to pomocí vztahu

$$V(t) = s_a(t) \exp(-j\omega_c t), \quad (4.8)$$

kde ω_c je nosný kmitočet

Spektrum komplexní obálky je zobrazeno na Obr. 8.



Obr. 8 Spektrum komplexní obálky signálu

Vstupní signál s přihlédnutím na vztah (4.8) lze pak přepsat do tvaru

$$s(t) = \text{Re}\{V(t)\exp(j\omega_c t)\}. \quad (4.9)$$

Komplexní obálka signálu obsahuje všechny informace, které obsahuje vstupní signál, ale je nezávislá na nosné frekvenci. Tato transformace signálu je výhodná především z důvodu absence rychlých obvodů, které by byly nutné v případě, že by signál obsahoval nosnou frekvenci. Veškeré zpracování signálů je tak možné provádět v základním pásmu [39].

4.3 Definice optimálního a přizpůsobeného filtru

Detekce cíle v přijatém signálu a odhad jeho parametrů (nejčastěji pozice a rychlost cíle) budou ovlivněny přítomností šumu. Pravděpodobnost detekce a přesnosti charakteristik cíle se budou zvyšovat s rostoucím poměrem odstupu signálu od šumu SNR . Optimální filtr je zařízení, které umožní pro daný vstupní signál maximalizovat parametr SNR [38].

Pro stanovení parametrů optimálního filtru je nutné stanovit komplexní frekvenční odezvu filtru $H(\omega)$, kde jediným kritériem pro určení $H(\omega)$ bude maximalizace SNR

Výstup ve frekvenční a časové oblasti po filtraci signálu bude dán vztahem

$$\begin{aligned} S_{out}(\omega) &= S(\omega)H(\omega) \\ s_{out}(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega)H(\omega)\exp(j\omega t)d\omega \end{aligned} \quad (4.10)$$

Výstupní signál dosáhne maximální hodnoty v určitém čase t_0 , poměr výkonu signálu k výkonu šumu v tomto maximu bude dán vztahem

$$\frac{P_s}{P_N} = \frac{1}{2\pi} \frac{\left| \int_{-\infty}^{\infty} \left(S(\omega) / \sqrt{S_N(\omega)} \right) \sqrt{S_N(\omega)} K(\omega) \exp(j\omega t_0) d\omega \right|^2}{\int_{-\infty}^{\infty} S_N(\omega) K^2(\omega) d\omega}. \quad (4.11)$$

Použijeme-li na výraz (4.11) Cauchy-Schwarzovu nerovnost, vztah přejde na tvar

$$SNR = \frac{P_s}{P_N} \leq \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{|S(\omega)|^2}{S_N(\omega)} d\omega. \quad (4.12)$$

Maximum nastane, pokud se pravá strana bude rovnat straně levé. Úpravou vztahu (4.10) a (4.12) dostaneme výsledný vztah pro frekvenční odezvu optimálního filtru danou výrazem

$$H(\omega) = \frac{AS(\omega)^*}{S_n(\omega)} \exp(-j\omega t_0), \quad (4.13)$$

kde A je konstanta.

Přizpůsobený filtr pak bude definován jako filtr pro maximální odstup SNR , kde šum bude pro většinu případů definován bílým šumem se spektrální hustotou $S_n(\omega) = \frac{N_0}{2}$.

Frekvenční odezva přizpůsobeného filtru pak bude po úpravě dána vztahem

$$H(\omega) = S(\omega)^* \exp(-j\omega t_0). \quad (4.14)$$

Impulsní odezvu získáme zpětnou Fourierovou transformací vztahu (4.14), kterou získáme zápisem

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega)^* \exp[-j\omega(t_0 - t)] d\omega = s(t_0 - t)^*, \quad (4.15)$$

kde $s(t)$ je vstupní signál. Přihlédneme-li k faktu, že vstupní signál bude splňovat podmínku $t_0 - t \geq 0$ (podmínka je splněna pro každý reálný signál), pak ze vztahu (4.15) lze odvodit vztah pro výstupní napětí, který bude dán vztahem

$$s_{out}(t) = \int_0^t s(x) h(t-x) dx, \quad (4.16)$$

kde $s(t)$ je vstupní signál, $h(t)$ je impulzní odezva přizpůsobeného filtru a $s_{out}(t)$ je výstupní signál.

Definice přizpůsobeného filtru bude dále využita v kapitole pojednávající o vzájemné funkci neurčitosti.

5 VZÁJEMNÁ FUNKCE NEURČITOSTI

Tato kapitola je věnována stěžejnímu tématu této práce, a to vzájemné funkci neurčitosti. Je členěna do několika navazujících podkapitol. První obsahuje matematické odvození modelů přímého a odraženého signálu s podrobným popisem jejich vlastností a dále následné vyjádření odraženého signálu, obsahujícího informaci o cíli, pomocí přímého signálu ve formátu vhodném pro další zpracování (vlastní výpočet CAF).

Další podkapitola je věnována již vlastnímu odvození vzájemné funkce neurčitosti, a to ve spojitě i diskrétní oblasti, s uvedením jejich vlastností. Součástí diskrétního tvaru CAF je i stanovení rozlišení v čase (vzdálenosti) i Dopplerově posunu, které je stěžejní pro stanovení maxima, a tím i pro přesné určení polohy cíle.

V dalších podkapitolách jsou postupně popsány vlastnosti vzájemné funkce neurčitosti a vlastní metody výpočtu CAF.

Princip detekce je založený na vzájemné korelaci mezi přímým a odraženým signálem u bistatického trojúhelníku (Obr. 1) pomocí vzájemné funkce neurčitosti. Počet bistatických trojúhelníků od jednoho cíle je dán počtem využitých příležitostných vysílačů (TO). Odražený signál má obdobný charakter jako přímý signál s tím rozdílem, že:

- Má větší zpoždění než přímý signál
- vlivem pohybu cíle dochází k posunu tzv. Dopplerovu posunu frekvence
- má o několik řádů nižší úroveň signálu (v porovnání s přímým signálem se jeho úroveň pohybuje kolem -120 dB)
- jeho průběh může být ovlivněn dalšími vlivy (zejména pozemním clutterem a efekty šíření podél povrchu Země)

Princip výpočtu CAF:

Přijatý odražený signál je dále posouván v čase o konstantní časovou hodnotu a s konstantním Dopplerovým posunem vzhledem k přímému signálu. Při každém z těchto časových posunů jsou vzorky mezi posunutým a konstantním (přímým) signálem vynásobeny a sčítány dohromady. Výsledkem těchto matematických operací přes celou délku signálu v čase bude křivka, v jejímž maximu si jsou signály „nejvíce podobné“. Toto maximum bude odpovídat určité hodnotě zpoždění τ_D .

Do této doby byl uvažován pouze posun v čase mezi přímým a zpožděným signálem s konstantní hodnotou Dopplerovského posunu. Tyto dva signály lze také posouvat

z hlediska Dopplerovských posunů v určitém intervalu při konstantním (neměnném) zpoždění τ_D . Vzájemným vynásobením a následnou sumací vznikne opět křivka, v jejímž maximu jsou si signály „nejvíce podobné“ pro odpovídající Dopplerovský posun ω_D .

Uvažujeme-li oba dva výše uvedené posuny (τ_D a ω_D) najednou, výsledkem bude plocha, v jejímž maximu leží (či spíše může ležet) hledaný cíl o odpovídajících hodnotách časového a Dopplerovského zpoždění. Otázka vzniku falešných cílů a jejich následné eliminaci, bude uvedena později.

Další úvahy budou popsány pro jednu dvojici vysílač - přijímač. Obdobné úvahy by platily i pro další dvojice, došlo by pouze k přeindexování signálů, viz Obr. 3. Signál přijatý přijímačem bude mít následující vybrané vlastnosti:

- časově omezený signál
- reálný signál (výkonové spektrum signálu se nachází jak v kladných, tak i záporných hodnotách, zrcadlově otočené)

5.1 Odvození modelů signálů pro výpočet CAF

Vysílaný signál $s_{T_0}(t)$ i oba přijímané signály a to přímý signál $s_T(t)$ a odražený signál $s_R(t)$ jsou pásmové signály, pro něž lze další postup nejjednodušeji popsat v komplexní reprezentaci. Proto zde budeme pracovat pouze s jejich analytickými signály a komplexními obálkami.

$$\begin{aligned}\hat{s}_{T_0}(t) &= V_{T_0}(t) \exp(j\omega_c t) \\ \hat{s}_T(t) &= V_T(t) \exp(j\omega_c t) \quad , \\ \hat{s}_R(t) &= V_R(t) \exp(j\omega_c t)\end{aligned}\tag{5.1}$$

kde: $\hat{s}_{T_0}(t)$, $\hat{s}_T(t)$, $\hat{s}_R(t)$ jsou analytické signály vysílaného (T_0), přímého (T) a odraženého (R) signálu

$V_{T_0}(t)$, $V_T(t)$, $V_R(t)$ jsou komplexní obálky těchto signálů

ω_c je nosný kmitočet signálů

Ve skutečnosti jsou oba přijímané signály kontaminovány ještě šumem, případně odrazy od pozemních objektů (clutterem). Pro účely této práce není přítomnost tohoto rušení v přijímaných signálech podstatná.

Signály přímý i odražený jsou zpožděnými a zeslabenými replikami vyslaného signálu:

$$\hat{s}_T(t) = a_T \hat{s}_{T0}(t - \tau_L)$$

$$\hat{s}_R(t) = a_R \hat{s}_{T0}(t - \tau_R) \Rightarrow \hat{s}_R(t) = \frac{a_R}{a_T} \hat{s}_T(t - \tau_R + \tau_L) = a \hat{s}_T(t - \tau_R + \tau_L) = a \hat{s}_T(t - \tau_D), \quad (5.2)$$

$$V_R(t) = a V_T(t - \tau_D) \exp(-j\omega_c \tau_D)$$

kde: τ_L , τ_R jsou zpoždění přímého a odraženého signálu oproti vyslanému signálu

τ_D je zpoždění odraženého signálu oproti přímému signálu

a_T , a_R jsou relativní změny (poklesy) amplitud přímého a odraženého signálu oproti vyslanému signálu

$a = \frac{a_R}{a_T}$ je relativní změna amplitudy odraženého signálu vůči přímému.

Zpoždění $\tau_D(t)$ je časově závislá veličina, kterou lze vyjádřit pomocí vztahu

$$\tau_D(t) = \frac{R_T(t) + R_R(t) - L}{c}, \text{ kde } c \text{ je rychlost světla, } R_T(t) \text{ je vzdálenost vysílač – cíl, } R_R(t)$$

je vzdálenost cíl – přijímač a L je vzdálenost přijímač – vysílač, kterou uvažujeme neměnnou. Označíme-li bistatickou vzdálenost $D(t) = R_T(t) + R_R(t)$ dostaneme

předchozí vztah ve tvaru $\tau_D(t) = \frac{D(t)}{c} - \frac{L}{c}$. Bistatická vzdálenost bude obecně dána

rozvojem $D(t)$ do Taylorovy řady ve tvaru

$$D(t) = D_0 + vt + a \frac{t^2}{2} + b \frac{t^3}{6} + c \frac{t^4}{24} \dots, \quad (5.3)$$

kde: v , a , b , c , ... jsou konstanty (v čase), závislé na poloze cíle v okamžiku $t = 0$.

Pokud je časový interval, který je k dispozici pro určení polohy cíle dostatečně krátký (cca do 1 s), lze považovat změnu polohy cíle na elipsoidu za zanedbatelnou. Za tohoto předpokladu lze členy s vyššími mocninami t zanedbat a výše uvedený vztah přejde na tvar:

$$D(t) = D_0 + vt. \quad (5.4)$$

Po dosazení do vztahu pro odražený signál dostaneme tvar:

$$\hat{s}_R(t) = a \hat{s}_T \left(t - \frac{D_0 + vt}{c} + \frac{L}{c} \right) = a \hat{s}_T \left[t \left(1 - \frac{v}{c} \right) - \frac{D_0 - L}{c} \right] = a \hat{s}_T \left[t \left(1 - \frac{v}{c} \right) - \tau_{D0} \right]. \quad (5.5)$$

Hodnota v nám vyjadřuje velikost Dopplerovské rychlosti v okamžiku $t=0$. Člen $\tau_{D0} = \frac{D_0 - L}{c}$ nám určuje časové zpoždění a D_0 je bistatická vzdálenost, (obojí v okamžiku $t=0$), jejíž velikost lze (teoreticky) určit pomocí následujícího vztahu:

$$D_0 = \sqrt{(x_{Tg} - x_T)^2 + (y_{Tg} - y_T)^2 + (z_{Tg} - z_T)^2} + \sqrt{(x_R - x_{Tg})^2 + (y_R - y_{Tg})^2 + (z_R - z_{Tg})^2}, \quad (5.6)$$

kde x_T, y_T, z_T a x_R, y_R, z_R jsou souřadnice vysílače a přijímače a x_{Tg}, y_{Tg}, z_{Tg} jsou souřadnice cíle v okamžiku $t=0$. Stanovení vzdálenosti D_0 je podle výše uvedeného vzorce již na první pohled nemožné z důvodu neznalosti souřadnic hledaného cíle.

Dosadíme-li vztah (5.5) pro odražený signál do vztahu (5.2), dostaneme následující vztah pro komplexní amplitudu odraženého signálu odraženého signálu

$$\begin{aligned} V_R(t) &= a.V_T \left[t \left(1 - \frac{v}{c} \right) - \tau_{D0} \right] \exp \left(-j\omega_c \left(\frac{v}{c} t + \tau_{D0} \right) \right) = \\ &= a.V_T \left[t \left(1 - \frac{v}{c} \right) - \tau_{D0} \right] \exp(j\omega_d t) \exp(-j\omega_c \tau_{D0}) \end{aligned}, \quad (5.7)$$

kde $\omega_d = -\omega_c \frac{v}{c}$ je Dopplerův posun kmitočtu.

Uvedený vztah lze zjednodušit za předpokladu, že ve výrazu $V_T \left[t \left(1 - \frac{v}{c} \right) - \tau_{D0} \right]$ lze

zanedbat člen $t \frac{v}{c}$, tedy když: $t \frac{v}{c} \ll \frac{1}{B}$, kde B je šířka pásma signálu. Zjednodušení je promítnuto do vztahu (5.8) a lze ho použít pouze v případě, že vstupní signál je úzkopásmový. Pro širokopásmové signály nelze zjednodušení použít.

Pro analytické signály:

$$\hat{s}_R(t) = a.\hat{s}_T \left[t \left(1 - \frac{v}{c} \right) - \tau_{D0} \right] \cong a.V_T(t - \tau_{D0}) \exp(j\omega_d t) \exp(-j\omega_c \tau_{D0}) \exp(j\omega_c t). \quad (5.8)$$

Pro komplexní obálky:

$$V_R(t) \cong a.V_T(t - \tau_{D0}) \exp(j\omega_d t) \exp(-j\omega_c \tau_{D0}). \quad (5.9)$$

Analyzujeme jednotlivé členy obsažené ve vztazích (5.8) a (5.9):

$V_T(t - \tau_{D0})$ je komplexní obálka vysílaného signálu časově posunutá o časové zpoždění τ_{D0}

$\exp(-j\omega_c\tau_{D0}) \equiv \exp(j\beta)$ je výraz vyjadřující konstantní fázový posun mezi odraženým a přímým signálem

$\exp(j\omega_c t)$ je nosná

$\exp(j\omega_d t)$ je výraz, vyjadřující Dopplerovský posun nosné

5.2 Definice vzájemné funkce neurčitosti

Vydeme z definice vzájemné funkce neurčitosti [40], [41], [42], [43] pro energetické analytické signály \hat{s}_T a \hat{s}_R :

$$CAF(\tau, \omega) = \int_{-\infty}^{+\infty} \hat{s}_T^*(t) \hat{s}_R(t + \tau) \exp(-j\omega t) dt. \quad (5.10)$$

Za odražený signál \hat{s}_R dosadíme vyjádření pomocí ze vztahu (5.8) a za přímý signál \hat{s}_T ze vztahu (5.1):

$$\begin{aligned} CAF(\tau, \omega) &= a \int_{-\infty}^{+\infty} V_T^*(t) \exp(-j\omega_c t) V_T(t - \tau_{D0} + \tau) \exp[j\omega_c(t - \tau_{D0} + \tau)] \cdot \\ &\cdot \exp(j\beta) \exp[j\omega_d(t - \tau_{D0} + \tau)] \exp(-j\omega t) dt = \\ &= \exp[j\omega_c(-\tau_{D0} + \tau)] \int_{-\infty}^{+\infty} V_T^*(t) V_R(t + \tau) \exp[j(\omega_d - \omega)t] dt = \\ &= a \exp(j\beta) \exp[j\omega_c(-\tau_{D0} + \tau)] \int_{-\infty}^{+\infty} V_T^*(t) V_T(t - \tau_{D0} + \tau) \exp[j(\omega_d - \omega)t] dt \end{aligned} \quad (5.11)$$

Absolutní hodnota CAF bude rovna:

$$\begin{aligned} |CAF(\tau, \omega)| &= \left| \int_{-\infty}^{+\infty} V_T^*(t) V_R(t + \tau) \exp[j(\omega_d - \omega)t] dt \right| = \\ &= |a| \left| \int_{-\infty}^{+\infty} V_T^*(t) V_T(t - \tau_{D0} + \tau) \exp[j(\omega_d - \omega)t] dt \right| \end{aligned} \quad (5.12)$$

Použijeme-li známou Cauchy – Schwarzovu nerovnost: $\left| \int_a^b s(x) dx \right| \leq \int_a^b |s(x)| dx$.

Dostaneme: $|CAF(\tau, \omega)| \leq |a| \int_{-\infty}^{+\infty} |V_T^*(t) V_T(t - \tau_{D0} + \tau) \exp[j(\omega_d - \omega)t]| dt$

Rovnost zřejmě nastane pouze při $\tau - \tau_{D0} = 0$ a $\omega_d - \omega = 0$:

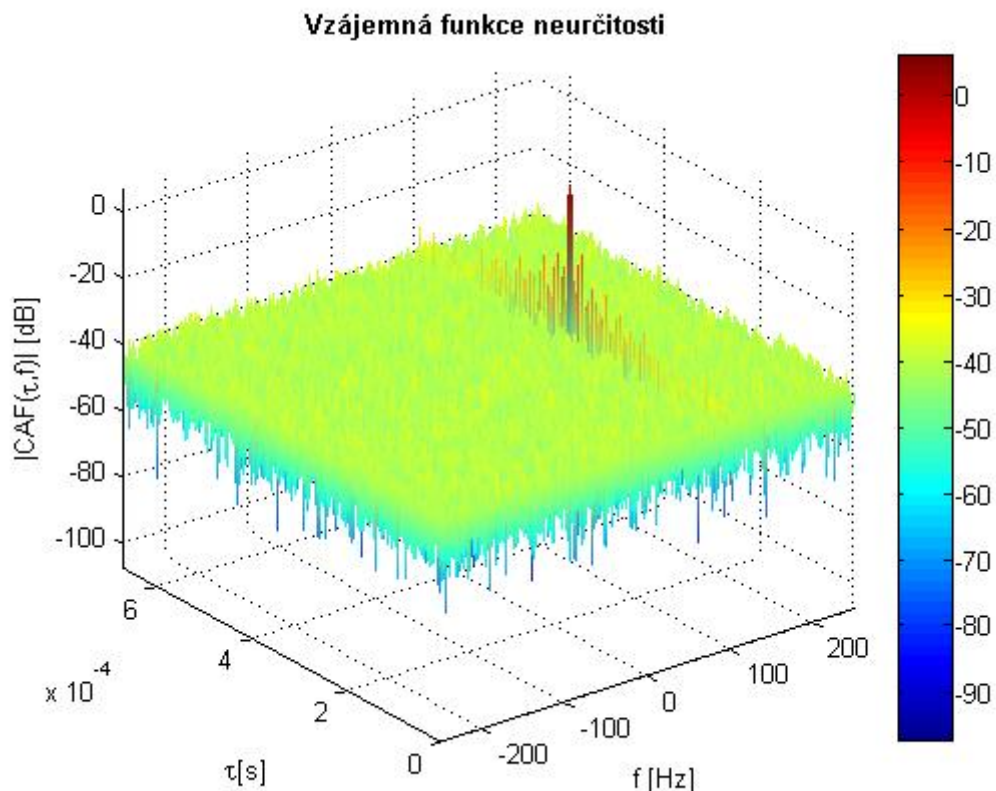
$$\left| \int_{-\infty}^{+\infty} V_T^*(t) V_T(t) dt \right| = \int_{-\infty}^{+\infty} |V_T^*(t) V_T(t)| dt,$$

takže absolutní hodnota této funkce dosahuje maxima, pokud $\tau = \tau_{D0}$ a $\omega_d = \omega$. Naším úkolem je tedy funkci $|CAF(\tau, \omega)|$ odhadnout a nalézt souřadnice jejího maxima. Přitom k jejímu výpočtu můžeme použít komplexní obálky přijatých signálů: $V_T(t)$ a $V_R(t)$ místo analytických signálů $\hat{s}_T(t)$ a $\hat{s}_R(t)$. V dalším pro zjednodušení zaměníme označení komplexních obálek za označení signálů takto: $V_T(t) \rightarrow s_T(t)$ a $V_R(t) \rightarrow s_R(t)$. Z praktických důvodů přejdeme ještě od proměnných ω a ω_D k frekvencím f a f_D .

Hledáme tedy maximum absolutní hodnoty funkce:

$$CAF(\tau, f) = \int_{-\infty}^{+\infty} s_T^*(t) s_R(t + \tau) \exp(-j2\pi ft) dt. \quad (5.13)$$

Ukázka vypočteného povrchu $|CAF(\tau, f)|$ je na Obr. 9.



Obr. 9 Ukázka vzájemné funkce neurčitosti

5.3 Vlastnosti CAF

V této kapitole budou uvedeny hlavní vlastnosti vzájemné funkce neurčitosti, které jsou založeny na předpokladu, že energie přímého a odraženého signálu jsou normalizované k jedné:

$$\int_{-\infty}^{+\infty} s_T^*(t) s_T(t) dt = \int_{-\infty}^{+\infty} s_R^*(t) s_R(t) dt = 1. \quad (5.14)$$

5.3.1 Maximum CAF

Absolutní hodnota $CAF(\tau, f)$ dosahuje maxima, jak již bylo řečeno při $\tau = \tau_{D0}$ a $f = f_{D0}$. V případě normalizovaných signálů bude toto maximum rovno jedné:

$$\begin{aligned} |CAF(\tau_{D0}, f_D)| &= \left| \int_{-\infty}^{+\infty} s_T^*(t) s_R(t + \tau_{D0}) \exp(j\beta) dt \right| = \int_{-\infty}^{+\infty} |s_T^*(t) s_R(t + \tau_{D0})| dt = \\ &= \int_{-\infty}^{+\infty} s_T^*(t) s_T(t) dt = 1 \end{aligned} \quad (5.15)$$

5.3.2 Konstantní objem CAF

Objem CAF je definován vztahem

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} |CAF(\tau, f)|^2 d\tau df = 1. \quad (5.16)$$

5.3.3 Symetrie CAF

Další vlastností CAF je symetrie k maximu funkce daná vztahem:

$$\begin{aligned} |CAF(\tau - \tau_{D0}, f - f_D)| &= \left| \int_{-\infty}^{+\infty} s_T^*(t) s_R(t + \tau - \tau_{D0}) \exp[j(\omega_d - \omega)t] dt \right| = \\ &= \left| \int_{-\infty}^{+\infty} s_T^*(t) s_T(t - 2\tau_{D0} + \tau) \exp[j(2\omega_d - \omega)t] dt \right| = \\ &= \left| \int_{-\infty}^{+\infty} s_T^*(t + 2\tau_{D0} - \tau) s_T(t) \exp[j(2\omega_d - \omega)(t + 2\tau_{D0} - \tau)] dt \right| = \\ &= \left| \int_{-\infty}^{+\infty} s_R^*(t + \tau_{D0} - \tau) s_T(t) \exp[j(\omega - \omega_d)t] dt \right| = |CAF(\tau_{D0} - \tau, f_D - f)| \end{aligned} \quad (5.17)$$

5.3.4 Řez v maximu $|CAF|$ při konstantním kmitočtu $f = f_D$

Nejprve provedeme řez funkcí CAF v maximu její absolutní hodnoty při konstantní $f = f_D$:

$$CAF(\tau, f_D) = \int_{-\infty}^{+\infty} s_T^*(t) s_R(t + \tau) dt = \int_{-\infty}^{+\infty} s_T^*(t - \tau) s_R(t) dt .$$

V předchozích kapitolách došlo k odvození přizpůsobeného filtru, který, jak již bylo zmíněno, je filtrem pro maximální odstup SNR a je definován vztahem (4.16).

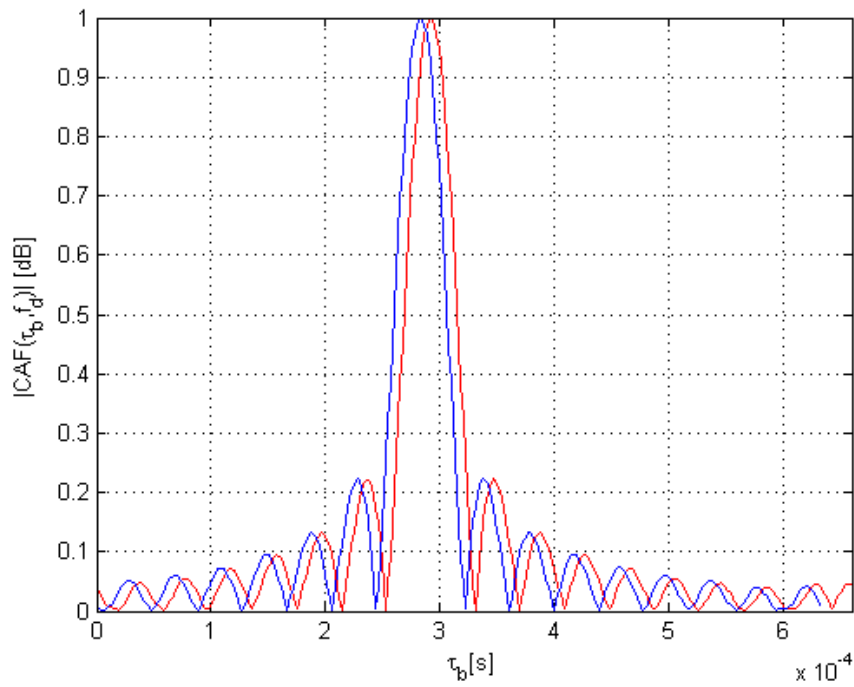
Dosadíme-li za vstupní signál $s(x)$ odražený signál $s_R(t)$ a za impulzní odezvu přizpůsobeného filtru $h(t-x)$ přímý signál $s_T^*(\tau-t)$ pak lze veličinu $CAF(\tau, f)$ při $f = f_D$ považovat za výstup přizpůsobeného filtru, takže výpočet funkce neurčitosti na kmitočtu Dopplerova posunu zjevně současně maximalizuje odstup signál – šum.

Matematické vyjádření absolutní hodnoty tohoto řezu lze popsat pomocí vztahu

$$|CAF(\tau, f_D)| = \int_0^T s_T(t) \overline{s_T(t - \tau_D + \tau)} \exp(j\beta) dt = |R(\tau)|, \quad (5.18)$$

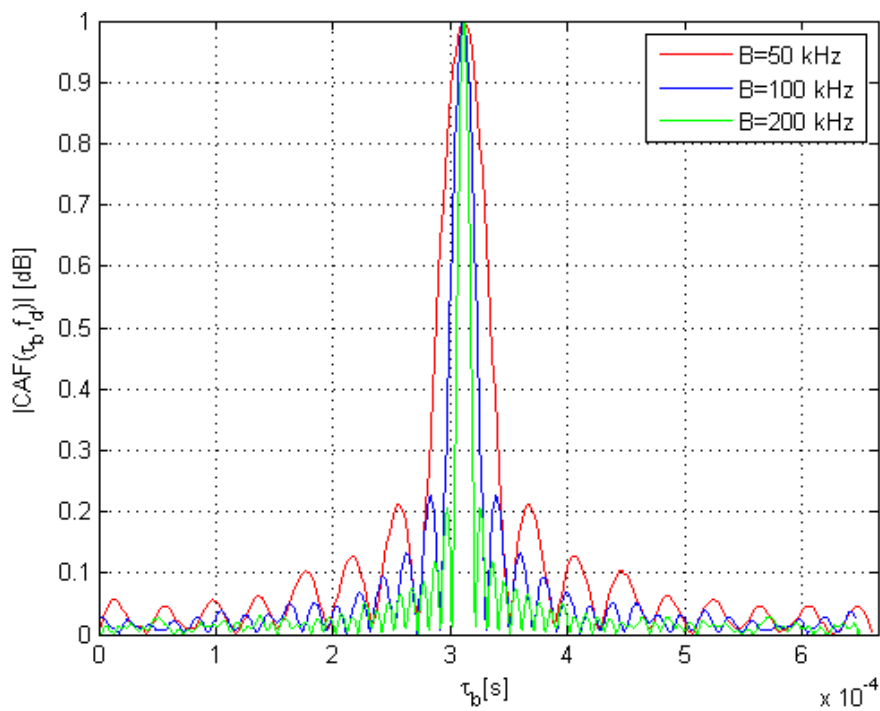
kde $|R(\tau)|$ vyjadřuje autokorelační funkci přímého i odraženého signálu.

Z této křivky lze na poklesu -3dB (čemuž odpovídá pokles na polovinu výkonu) určit šířku hlavního laloku, jejíž převrácená hodnota odpovídá šířce pásma vysílaného signálu. S šířkou hlavního laloku také úzce souvisí rozlišovací schopnost $\Delta\tau$ v čase, která je definována jako minimální časový odstup cílů, které je schopen radar od sebe odlišit. V případě, že šířka pásma vysílaného signálu bude nízká a cíle se budou pohybovat v těsné blízkosti, dojde k překryvu dvou „širokých“ laloků a vznikne jeden velký lalok, z kterého nepůjde rozpoznat, že se jedná o dva cíle. Ukázka překrytí laloků dvou cílů, umístěných v bistatických vzdálenostech $R_{B1} = 90km$ a $R_{B2} = 92km$, je na Obr. 10.



Obr. 10 Ukázka špatného rozlišení dvou cíů

Vliv šířky pásma vstupního signálu na šířku hlavního laloku pro hodnoty $B = 50\text{kHz}, 100\text{kHz}, 200\text{kHz}$ je zobrazen na Obr. 11.



Obr. 11 Vliv šířky pásma na šířku hlavního laloku $|CAF|$

5.3.5 Řez v maximu $|CAF|$ při konstantním zpoždění $\tau = \tau_{D0}$

Provedeme-li řez $CAF(\tau, f)$ v maximu její absolutní hodnoty při konstantním posuvu $\tau = \tau_D$, dostaneme po úpravě vyjádření CAF ve tvaru

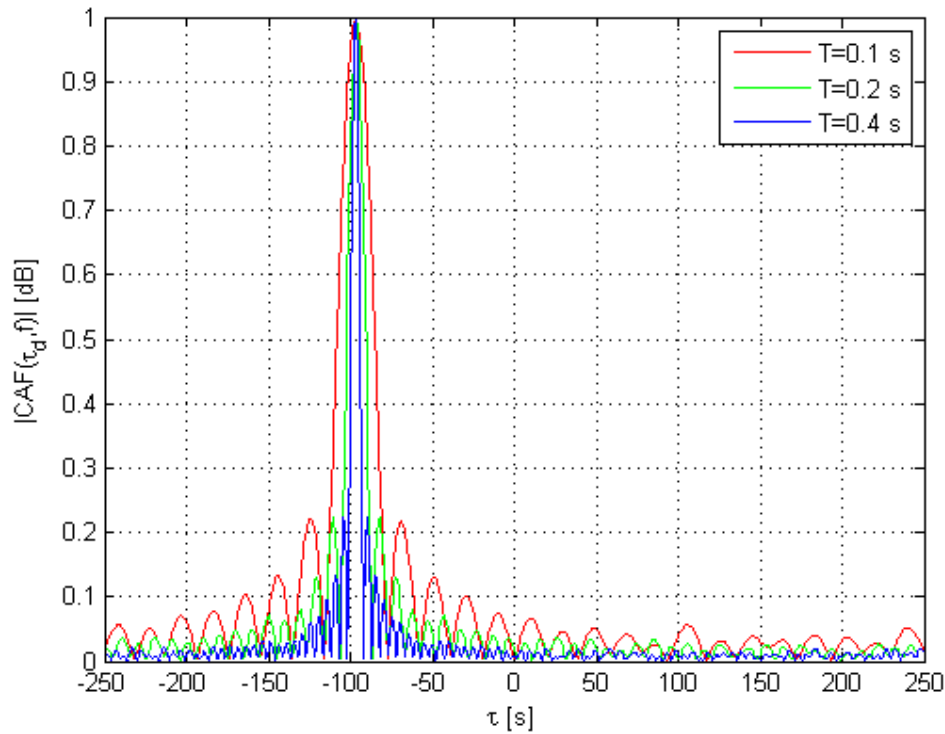
$$\begin{aligned} CAF(\tau_{D0}, f) &= \int_{-\infty}^{+\infty} |s_T(t)|^2 \exp(j\beta) \exp[-j2\pi(f - f_D)t] dt = \\ &= \int_{-\infty}^{+\infty} |s_T(t)|^2 \exp(j\beta) \exp(j2\pi f_D t) \exp(-j2\pi f t) dt = \quad , \quad (5.19) \\ &= \mathbf{F} \left[|s_T(t)|^2 \exp(j\beta) \exp(j2\pi f_D t) \right] = \mathbf{F}[\gamma(t)] \end{aligned}$$

kde $\gamma(t) = |s_T(t)|^2 \exp(j\beta) \exp(j2\pi f_D t)$.

Vztah (5.19) nám vyjadřuje Fourierovu transformaci funkce $\gamma(t)$, která je fázově a frekvenčně posunutou verzí $|s_T(t)|^2$.

Doposud jsme předpokládali, že signály $s_T(t)$ a $s_R(t)$ jsou energetické. Ve skutečnosti jsou rozhlasové, televizní a další signály, používané pro PCL signály výkonové. Pro zpracování však vždy používáme pouze konečné úseky těchto signálů, takže z nich vlastně uděláme signály finitní (tedy také energetické), nenulové v nějakém konečném intervalu například $t \in \langle 0, T \rangle$. Dále budeme tedy signály $s_T(t)$ a $s_R(t)$ považovat za finitní, nenulové v intervalu délky T .

Vzhledem k vlastnostem Fourierovy transformace (věta o změně měřítka) bude převrácená hodnota šířky hlavního laloku $|CAF(\tau_{D0}, f)|$ přímo úměrná délce intervalu T . Minimální odstup Dopplerových kmitočtů Δf_D dvou cílů, které ještě lze rozeznat se nazývá rozlišovací schopnost radaru v Dopplerově kmitočtu. Za rozlišovací schopnost Δf_D se považuje šířka hlavního laloku $|CAF|$ v řezu $\tau = \tau_{D0}$. Bude-li integrační doba T delší, pak šířka hlavního laloku bude užší a maximum výraznější. Obecně lze říci, že v praxi je požadavek na co nejdelší dobu integrace z důvodu užší šířky hlavního laloku a tím vyššího rozlišení v Doppleru (obdobně jak tomu bylo u rozlišení v čase, viz Obr. 10). Zároveň však delší doba integrace způsobí velký nárůst výpočetních operací a tím delší dobu výpočtu, což je nežádoucí. Doba integrace je omezena také pohybem cíle a tím změnou polohy maxima $|CAF|$ za dobu integrace. Optimální doba integrace vzhledem k době zpracování dat a dostatečného rozlišení v Doppleru bývá obvykle v intervalu 0,6 – 1,2 s. Vliv doby integrace na šířku hlavního laloku je zobrazen na Obr. 12.



Obr. 12 Řez CAF v maximu při konstantním časového posuvu $\tau = \tau_{D0}$

Vzájemnou funkci neurčitosti $CAF(\tau, f)$ lze transformovat na tvar $CAF(R_B, f)$, kde $R_B = t \cdot c$ je bistatická vzdálenost a c je rychlost šíření signálu. Vyjádříme-li vzájemnou funkci neurčitosti v závislosti na bistatické vzdálenosti a Dopplerově posuvu dostaneme

$$\begin{aligned}
 CAF(R_B, f) &= CAF(c \cdot \tau, f) = \\
 &= \int_0^T s_T(t) \overline{s_T(t - \tau_D + \tau)} \exp(j\beta) \exp[-j2\pi(f - f_D)t] dt
 \end{aligned} \tag{5.20}$$

Dále bude v této práci vždy použito zobrazení $CAF(R_B, f)$ místo $CAF(\tau, f)$. Zobrazený povrch bude shodný s Obr. 9, pouze dojde ke změně osy Y.

Vztah (5.12) a (5.20) určuje průběh CAF ve spojitém čase, který pro výpočet nelze použít z důvodu nekonečného počtu hodnot ke zpracování. Výsledkem výpočtu ve spojitém čase by byla spojitá plocha v 3D prostoru, v jejímž vrcholu (vrcholech) se vyskytoval možný cíl (cíle) v odpovídajících bistatických vzdálenostech a Dopplerovských posunech. V praxi je CAF vypočtena pouze v diskrétních hodnotách, a to jak v časovém zpoždění, tak i v Dopplerově posuvu. Stanovení rozlišení CAF, bude uvedeno v kapitole 5.4 současně s diskrétním tvarem vzájemné funkce neurčitosti.

5.4 CAF v diskretní oblasti

Odvození diskretního tvaru CAF je klíčové pro numerický výpočet CAF používaných v radarových systémech. V této kapitole bude provedeno jeho odvození s uvedením parametrů, které určují vlastnosti diskretní verze CAF .

Do výpočtu $CAF(\tau, f)$ vstupují komplexní obálky přímého a odraženého signálu $s_T(t)$ a $s_R(t)$, definované v intervalu $t \in \langle 0, T \rangle$. V diskretním čase jsou to vzorky těchto signálů:

$$\begin{aligned} s_T(t_n) &\equiv s_T[n] \\ s_R(t_n) &\equiv s_R[n] \end{aligned} \quad (5.21)$$

v okamžicích $t_n = k.T_S = n / f_s$,

kde: T_S je vzorkovací perioda,

$f_s \geq 2B$ je vzorkovací kmitočet,

B je šířka pásma $s_T(t)$ a $s_R(t)$.

Mezi výchozí parametry tedy patří délka signálu T a vzorkovací frekvence signálu f_s (tyto parametry budou společné pro přímý i odražený signál). Místo proměnné τ se často v těchto aplikacích používá proměnná $R_B = ct = D - L$ (kde D je bistatická vzdálenost a L je vzdálenost mezi vysílačem a přijímačem, c je rychlost šíření signálu).

Celkový počet vzorků každého ze signálů bude roven $N = f_s T$.

Jak již bylo řečeno, výsledkem výpočtu $|CAF|$ ve spojitě oblasti by byla spojitá plocha, jejíž hranice by byly ohraničeny maximální velikostí časového zpoždění τ_{\max} (nebo maximální délkou bistatické vzdálenosti $R_{B\max}$) a maximální velikostí Dopplerova posunu f_{\max} .

Velikost kroku v bistatické vzdálenosti (v angl. *range bin*), kterou označíme dR je rovna:

$$dR = \frac{c}{f_s}, \quad (5.22)$$

Velikost kroku v Dopplerově posunu, kterou označíme dF je rovna:

$$dF = \frac{f_s}{N}. \quad (5.23)$$

Celkový počet kroků ve vzdálenosti R_B a v Doppleru f , pak bude určen velikostí maximální bistatické vzdálenosti cíle R_{max} [m] a maximálního očekávaného Dopplerova posunu F_{max} [Hz]

$$\begin{aligned} R_{bin} &= \left\lceil \frac{R_{max}}{dR} \right\rceil \\ F_{bin} &= \left\lceil \frac{F_{max}}{dF} \right\rceil, \end{aligned} \quad (5.24)$$

kde R_{bin} je celkový počet kroků v bistatické vzdálenosti a F_{bin} je celkový počet kroků v Dopplerově posunu.

Pro vyjádření CAF v diskrétní oblasti vyjdeme ze vztahu (5.13), který po úpravě dostane tvar

$$\begin{aligned} CAF(\tau_m, f) &= \int_0^T s_T^*(t) s_R(t + \tau_m) \exp(-j2\pi ft) dt = \\ &= \int_0^T r_m(t) \exp(-j2\pi ft) dt = \mathbf{F}[r_m(t)] \end{aligned} \quad (5.25)$$

kde $\{\tau_m\}$, pro $m = 1$ až R_{bin} je množina hodnot očekávaných zpoždění, $r_m(t)$ je vynásobený přímý a zpožděný signál (posunutý jak časově, tak i frekvenčně), tzv. *lag product* a $\mathbf{F}[r_m(t)]$ je Fourierova transformace tohoto součinu. Výpočet CAF je tedy podle vztahu (5.25) definován pro každé časové zpoždění z množiny hodnot $\{\tau_m\}$ jako Fourierova transformace součinu přímého a odraženého signálu.

Převodem součinu přímého a odraženého signálu do diskrétní oblasti dostaneme

$$r_m[n] = s_T[n] s_R[n+m]. \quad (5.26)$$

Diskrétní tvar CAF bude s využitím vztahů (4.2) a (5.26) roven:

$$\begin{aligned} CAF(m, k) &= \sum_{n=0}^{N-1} s_T^*[n] s_R[n+m] \exp\left(\frac{-j2\pi kn}{N}\right) = \\ &= \sum_{n=0}^{N-1} s_1^*[n] s_2[n+m] \exp\left(\frac{-j2\pi kn}{N}\right), \end{aligned} \quad (5.27)$$

kde s_1 a s_2 je nové označení komplexních obálek přímého a odraženého signálu v diskrétním tvaru, k je pořadí Dopplerovského posunu $f_k = k/T$, m je pořadí časového zpoždění τ_m .

Pro hodnoty $m = m_0$ a $k = k_0$, pro něž časové zpoždění τ_m a Dopplerův posun f_k odpovídají časovému posunu τ_{D0} a Dopplerovu posunu f_D odraženého signálu $s_2[n]$ oproti přímému $s_1[n]$ bude $|CAF(m, k)|$ dosahovat svého maxima

$$|CAF(m, k)| = |CAF(m, k)|_{\max} \quad (5.28)$$

Teoretický počet bistatických vzdáleností $N_{\tau \max}$, v kterém lze vypočítat $CAF(m, k)$, je dán vztahem

$$N_{\tau \max} = 2N - 1, \quad (5.29)$$

který se v praxi v podstatě nevyužívá, a to kvůli tomu, že s_R je zpožděný oproti s_T a proto nemá smysl hledat maximum při záporných časových posunech. Dalším důvodem je omezený dosah radaru čemuž odpovídá omezení maximálního zpoždění odraženého signálu, které bude odpovídat pouze zlomku rozsahu celkového časového zpoždění $m \in \langle 0, N \rangle$. Počet vyhodnocovaných zpoždění N_τ pak bude dán nerovností $N_\tau \ll N_{\tau \max}$.

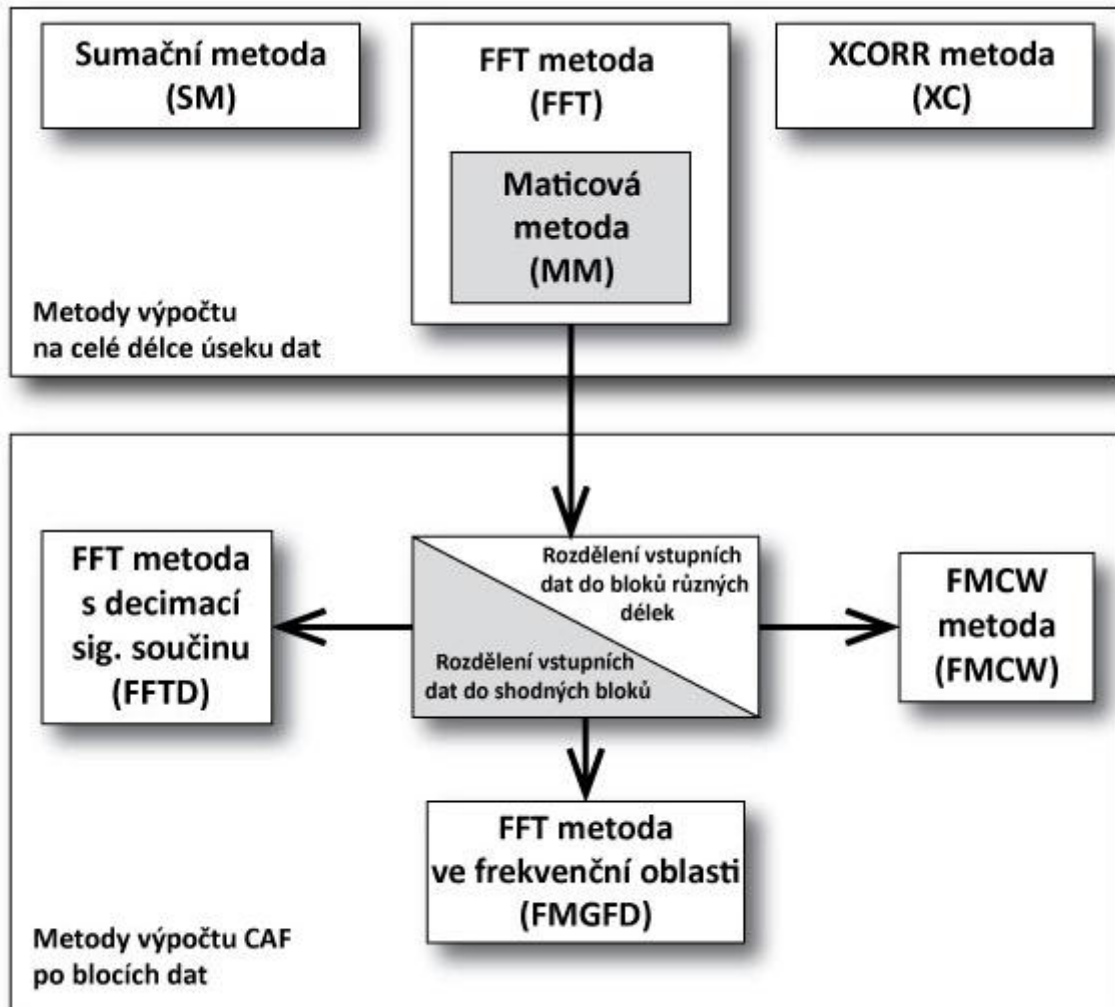
Obdobné odvození bude platit i pro počet Dopplerovských posunů. Maximální počet posunů $N_{k \max}$ bude roven $N_{k \max} = N + 1$, $k \in \langle -N/2, N/2 \rangle$. Skutečný počet Dopplerovských posunů N_k bude s ohledem na maximální rychlost cíle menší a lze psát $N_k \ll N_{k \max}$.

V této práci bude nadále uvažován vždy omezený rozsah bistatických vzdáleností i Dopplerovských posunů.

Hodnoty parametrů T , f_s , R_{\max} , F_{\max} atd. jsou uvedeny v kapitole 7.2.

5.5 Metody výpočtu diskrétního tvaru CAF

Vzájemnou funkci neurčitosti je možné vypočítat mnoha způsoby, které budou uvedeny v této kapitole. Jednotlivé metody výpočtu lze rozdělit do dvou základních skupin, a to metody bez dělení dat a metody s dělením dat do bloků. U každé metody bude uveden teoretický rozbor dané metody, výpis jejích vlastností a náročnosti výpočtu z hlediska počtu numerických operací za sekundu (FLOPS) a algoritmu výpočtu. Základní metody algoritmů výpočtu CAF vychází z prací p. Stein S., Winograd S., Tolimieri R., Auslander L. uvedených v [40], [41], [42], [43]. Schematické propojení jednotlivých metod je zobrazeno na Obr. 13.



Obr. 13 Schematické zobrazení vazeb mezi jednotlivými metodami

Metody výpočtu CAF na celé délce dat jsou založeny na přímém výpočtu CAF na základě vztahu (5.27). Jedná se o sumační metodu, korelační metodu, metodu využívající rychlou Fourierovu transformaci a maticovou metodu, která neobsahuje žádné cykly a je obzvláště vhodná pro rychlé výpočty CAF .

Druhá skupina metod je založena na rozdělení do bloků a výpočtu CAF blocích. Z důvodu opakujících se výpočtů přes jednotlivé bloky dochází k snížení výpočetní náročnosti a tím ke zkrácení výpočetní doby.

5.5.1 Sumační metoda (SM)

Sumační metoda vychází z přímého výpočtu CAF pomocí vztahu (5.27). Jedná se o základní metodu výpočtu, která nejvíce odpovídá definičnímu vztahu CAF a všechny ostatní metody budou vždy porovnávány s touto metodou. Algoritmus metody výpočtu je uveden v Tab. 2. Základní nevýhodou této metody je fakt, že algoritmus

metody používá dvě vnořené smyčky, které jsou velice numericky a časově náročné obzvláště pro velké N (celkový počet vzorků signálu). Výhodou této metody je možnost výpočtu CAF pro libovolné hodnoty m a k , což, jak uvidíme dále, nelze aplikovat u všech metod. Numerická náročnost metody (počet matematických operací) v požadovaném rozsahu bistatických vzdáleností a Dopplerovských posunů bude dána vztahem převzatým z [45]

$$FLOPS_{sumace} = N_r N_k N [FLOPS]. \quad (5.30)$$

Algoritmus sumační metody (SM)

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána

2. Algoritmus vlastního výpočtu CAF

for $m = 0$ do R_{bin}

for $k = 0$ do F_{bin}

$$CAF(m, k) = \sum_{n=0}^{N-1} r_m[n] \exp\left(\frac{-j2\pi kn}{N}\right), \text{ kde}$$

$$r_m[n] = s_T[n] s_R[n+m]$$

end;

end;

3. Stanovení maxima $CAF(m, k)$

a. Řez v maximu funkce $CAF(m, k_{max}) \rightarrow$ interpolace pro získání vyššího rozlišení ve vzdálenosti v rámci buňky, ve které se nachází cíl \rightarrow nalezení maxima interpolovaného průběhu m_{max}

b. Řez v maximu funkce $CAF(m_{max}, k) \rightarrow$ interpolace pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima k_{max}

Tab. 2. Algoritmus výpočtu CAF sumační metodou (SM)

5.5.2 Korelační metoda (XC)

Pro odvození korelační metody vyjdeme opět ze vztahu (5.27), který upravíme do tvaru korelační funkce definované vztahem

$$R_{xy}(\tau) = \sum_{n=0}^{N-1} x[n]y^*[n+\tau]. \quad (5.31)$$

Srovnáním vztahů (5.27) a (5.31) dostaneme upravený vztah pro výpočet *CAF*

$$\begin{aligned} CAF(m, k) &= \sum_{n=0}^{N-1} s_T^*[n] \left[s_R[n+m] \exp\left(\frac{j2\pi k(n+m)}{N}\right) \exp\left(\frac{-j2\pi km}{N}\right) \right] \\ CAF(m, k) &= \sum_{n=0}^{N-1} s_T^*[n] s_R[n+m] \exp\left(-j2\pi \frac{kn}{N}\right) = \sum_{n=0}^{N-1} \left\{ s_T^*[n] \exp\left(-j2\pi \frac{k}{N} n\right) \right\} s_R[n+m] = . \quad (5.32) \\ &= \sum_{n=0}^{N-1} x^*[n] y[n+m] = R_{xy}(m) \end{aligned}$$

První člen $\left\{ s_T^*[n] \exp\left(-j2\pi \frac{k}{N} n\right) \right\}$ vyjadřuje signál $x[n]$ a člen $\{s_R[n+m]\}$ vyjadřuje signál $y[n+m]$. Algoritmus výpočtu pracuje na následujícím principu. Pro jednotlivé Dopplerovy posuny jsou vypočítány $x[n]$ vzájemné korelační funkce $x[n]$ a $y[n]$ na celém rozsahu hodnot časových posunů $N_{\tau \max}$. Výhodou této metody je, že můžeme dosáhnout libovolného kroku v Doppleru, kde nejsou žádná omezení pro námi zvolenou oblast Dopplerových posunů. Numerická náročnost této metody je dána vztahem (viz [45]):

$$FLOP_{xcorr} = 3N_k N (2 + 5 \log_2 N) \text{ [FLOPS]}. \quad (5.33)$$

Algoritmus výpočtu vzájemné korelace, lze optimalizovat pomocí využití FFT. Korelace není počítána pomocí definičního vztahu (5.31), ale pomocí výpočtu korelace s využitím FFT výpočtu, kdy využijeme vlastnosti, že korelace ve frekvenční oblasti je definována prostým vynásobením spekter těchto signálů $X[p] = F[x(n)]$ a $Y[p] = F[y(n)]$

$$R_{xy}(\tau) = FFT^{-1} [F[x(n)]F[y(n)]] \equiv FFT^{-1} [FFT[x(n)]FFT^*[y(n)]]. \quad (5.34)$$

Tuto metodu výpočtu *CAF* je vhodné zvolit, pokud je rozsah očekávaných Dopplerových posunů N_k daleko menší, než je očekávaný rozsah časových zpoždění N_τ .

Algoritmus korelační metody XM :

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána.

2. Algoritmus vlastního výpočtu CAF

for $k = 0$ do F_{bin}

$$CAF_{full}(m, k) = \frac{1}{N} \sum_{n=0}^{N-1} \left\{ s_T^*[n] \exp\left(-j2\pi \frac{k}{N} n\right) \right\} s_R[n+m]$$

end;

3. Ořez funkce $CAF_{full}(m, k)$ vypočtené pro všechny N_{rmax} na požadovaný rozsah $CAF(m, k)$ pro $m \in \langle 0, R_{bin} \rangle$.

4. Stanovení maxima $CAF(m, k)$

a. Řez v maximu funkce $CAF(m, k_{max}) \rightarrow$ interpolace pro získání vyššího rozlišení ve vzdálenosti v rámci buňky, ve které se nachází cíl \rightarrow nalezení maxima interpolovaného průběhu m_{max}

b. Řez v maximu funkce $CAF(m_{max}, k) \rightarrow$ interpolace pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima k_{max}

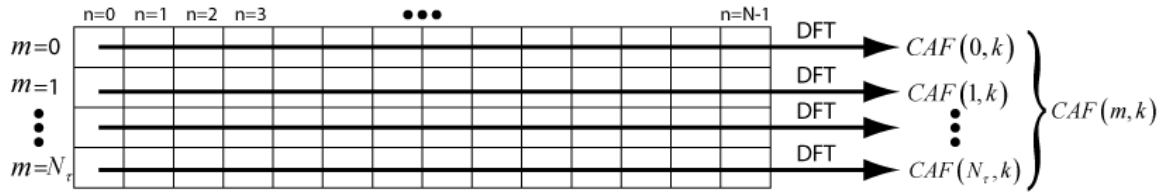
Tab. 3. Algoritmus výpočtu CAF korelační metodou (XC)

5.5.3 FFT metoda (FFT)

Pro odvození metody výpočtu, CAF založené na rychlé Fourierově transformaci FFT, vyjdeme z definice pro diskretní Fourierovu transformaci DFT definovanou vztahem (4.2). Nahradíme-li proměnnou s_k součinem přímého a odraženého signálu, vztah pro výpočet CAF dostane tvar

$$\begin{aligned} CAF(m, k) &= \sum_{n=0}^{N-1} s_T^*[n] s_R[n+m] \exp\left(\frac{-j2\pi kn}{N}\right) = \\ &= \sum_{n=0}^{N-1} r_m[n] \exp\left(\frac{-j2\pi kn}{N}\right) \equiv DFT[r_m[n]] \rightarrow FFT[r_m[n]] \end{aligned} \quad (5.35)$$

Pro výpočet DFT je běžně používána „rychlá“ forma výpočtu DFT, tzv. rychlá Fourierova transformace FFT [35], [36], [37]. Grafické zobrazení výpočtu je na Obr. 14.



Obr. 14 Grafické zobrazení výpočtu CAF pomocí metody FFT

Nevýhodou FFT metody je, že rozsah potřebných Dopplerových posunů daných množinou hodnot k je daleko menší, než rozsah hodnot, počítaných touto metodou (viz. kapitola 5.4). Většina výpočtů se tak provádí zbytečně a dochází k plýtvání výpočetní kapacitou. Celkový počet matematických operací na definovaném rozsahu je dán vztahem

$$P_{FFT} = 10N_r N \log_2 N \text{ [FLOPS]}. \quad (5.36)$$

Tuto metodu výpočtu CAF je vhodné zvolit, pokud je rozsah očekávaných časových zpoždění N_τ daleko menší, než je očekávaný rozsah Dopplerových posunů N_k .

Algoritmus FFT metody (FFT):

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána.
2. Algoritmus vlastního výpočtu CAF

for $k = 0$ do R_{bin}

vytvoření posunutého referenčního signálu $s_s[n]$ pro každou hodnotu R_{bin}

$$CAF_{full}(m, k) = \frac{1}{N} FFT[s_r^*[n]s_s[n+m]]$$

end;

3. Ořez funkce $CAF_{full}(m, k)$ vypočtené pro všechny $N_{k_{max}}$ na požadovaný rozsah $CAF(m, k)$ pro $k \in \langle 0, F_{bin} \rangle$

4. Stanovení maxima $CAF(m, k)$

a. Řez v maximu funkce $CAF(m, k_{max}) \rightarrow$ interpolace pro získání vyššího rozlišení ve vzdálenosti v rámci buňky, ve které se nachází cíl \rightarrow nalezení maxima interpolovaného průběhu m_{max}

b. Řez v maximu funkce $CAF(m_{max}, k) \rightarrow$ interpolace pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima k_{max}

Tab. 4. Algoritmus výpočtu CAF pomocí metody (FFT)

5.5.4 Maticová metoda výpočtu (MM)

Maticová metoda výpočtu CAF je v podstatě metoda FFT v maticovém tvaru. Umožňuje výpočet bez použití cyklů (smyček) a je proto obzvláště vhodná pro paralelizaci, která urychluje výpočty.

Předpokládejme, že máme k dispozici přímý signál $s_T[n]$ a odražený signál $s_R[n]$, které vytvářejí řádkové vektory: $\mathbf{s}_T \equiv \{s_T[n]\}$, $\mathbf{s}_R \equiv \{s_R[n]\}$ a časový řádkový vektor \mathbf{t} o N prvcích $t_n = \frac{n}{f_s}$.

Celková délka časového vektoru je dána vztahem $T = \frac{N}{f_s}$. Řádkový vektor

Dopplerových posunů označíme $\mathbf{F}_{\text{dop}} \equiv \left\{ f_k = \frac{k-1}{T} \right\}_1^{N_k}$, řádkový vektor časových posunů

bude: $\mathbf{T}_t \equiv \left\{ \tau_m = \frac{m-1}{f_s} \right\}_1^{N_t}$ a odpovídající vektor bistatických vzdáleností bude: $\mathbf{R}_b \equiv c \cdot \mathbf{T}_t$.

Frekvenční osa pro spektra signálů bude definována takto: $\mathbf{f}_{\text{osa}} \equiv \left\{ f_{or} = \frac{r-1}{T} \right\}_1^N$.

Algoritmus výpočtu je založen na vynásobení dvou matic \mathbf{M}_1 a \mathbf{M}_2 . $\mathbf{M}_1 \equiv \{s_R[n+m]\}$, kde n je řádek a m je sloupec. Každý sloupec této matice je tedy tvořen odraženým signálem $s_R[n+m]$, posunutým o τ_m . $\mathbf{M}_2 \equiv \left\{ s_T[n] \cdot \exp\left(\frac{j2\pi kn}{N}\right) \right\}$, kde je n řádek a k je sloupec. Každý sloupec této matice je tedy tvořen přímým signálem, frekvenčně posunutým, o f_k .

Jednotlivé kroky výpočtu \mathbf{M}_1 a \mathbf{M}_2 jsou následující:

- 1) Výpočet řádkového vektoru spektra odraženého signálu $\mathbf{S}_R = \mathbf{F}[\mathbf{s}_R]$, kde: $\mathbf{S}_R \equiv \{\mathbf{S}_R[r] = \mathbf{F}[s_R[n]]\}$
- 2) Vytvoření pomocné matice \mathbf{M}_{pom1} tvořenou N_t sloupci a N řádky:

$$\mathbf{M}_{\text{pom1}} = (\mathbf{S}_R)^T * \text{ones}(1, N_t) \equiv \underbrace{\begin{pmatrix} S_R(1) & S_R(1) & \dots & S_R(1) \\ S_R(2) & S_R(2) & \dots & S_R(2) \\ \vdots & \vdots & \vdots & \vdots \\ S_R(N) & S_R(N) & \dots & S_R(N) \end{pmatrix}}_{N_t \text{ - stejných sloupců}}, \quad (5.37)$$

kde: \mathbf{A}^T je matice (reálně) transponovaná k matici \mathbf{A} ,

* představuje maticové násobení.

- 3) Vytvoření matice \mathbf{M}_τ časových zpoždění ve spektrální oblasti o rozměru N řádků x N_τ sloupců:

$$\mathbf{M}_\tau = e^{[-j2\pi(\mathbf{f}_{osa})^T \cdot \mathbf{T}_\tau]} \equiv \underbrace{\begin{pmatrix} \exp(-j2\pi f_{osa1}\tau_1) & \exp(-j2\pi f_{osa1}\tau_2) & \dots & \exp(-j2\pi f_{osa1}\tau_{N_\tau}) \\ \exp(-j2\pi f_{osa2}\tau_1) & \exp(-j2\pi f_{osa2}\tau_2) & \dots & \exp(-j2\pi f_{osa2}\tau_{N_\tau}) \\ \vdots & \vdots & \dots & \vdots \\ \exp(-j2\pi f_{osaN}\tau_1) & \exp(-j2\pi f_{osaN}\tau_2) & \dots & \exp(-j2\pi f_{osaN}\tau_{N_\tau}) \end{pmatrix}}_{N_\tau \text{ sloupců}} \left. \vphantom{\begin{pmatrix} \exp(-j2\pi f_{osa1}\tau_1) \\ \exp(-j2\pi f_{osa2}\tau_1) \\ \vdots \\ \exp(-j2\pi f_{osaN}\tau_1) \end{pmatrix}} \right\} N \text{ řádků} \cdot \quad (5.38)$$

- 4) Matice \mathbf{M}_1 zpožděných replik odraženého signálu v časové oblasti pak bude dána vztahem:

$$\mathbf{M}_1 = F^{-1} [\mathbf{M}_{\text{pom1}} \cdot \mathbf{M}_\tau], \quad (5.39)$$

kde operátor \cdot označuje násobení prvek po prvku (dot product).

- 5) Vytvoření druhé pomocné matice \mathbf{M}_{pom2} dané replikací komplexně sdruženého přímého signálu $s_T^*[n]$:

$$\mathbf{M}_{\text{pom2}} = \left\| \begin{matrix} s_T^*(1) & s_T^*(2) & \dots & s_T^*(N) \\ s_T^*(1) & s_T^*(2) & \dots & s_T^*(N) \\ \vdots & \vdots & \ddots & \vdots \\ s_T^*(1) & s_T^*(2) & \dots & s_T^*(N) \end{matrix} \right\|_{N_k \text{ stejných řádků}} \cdot \quad (5.40)$$

- 6) Vytvoření matice Dopplerovských posunů z požadovaného rozsahu hodnot

$$\mathbf{M}_{\text{Dop}} = e^{[j2\pi(\mathbf{F}_{\text{Dop}})^T \cdot \mathbf{t}]^*} \equiv \underbrace{\left\| \begin{matrix} \exp(-j2\pi f_1 t_1) & \exp(-j2\pi f_2 t_1) & \dots & \exp(-j2\pi f_1 t_N) \\ \exp(-j2\pi f_2 t_1) & \exp(-j2\pi f_2 t_2) & \dots & \exp(-j2\pi f_2 t_N) \\ \vdots & \vdots & \ddots & \vdots \\ \exp(-j2\pi f_{N_k} t_1) & \exp(-j2\pi f_{N_k} t_2) & \dots & \exp(-j2\pi f_{N_k} t_N) \end{matrix} \right\|}_{N \text{ sloupců}} \left. \vphantom{\left\| \begin{matrix} \exp(-j2\pi f_1 t_1) \\ \exp(-j2\pi f_2 t_1) \\ \vdots \\ \exp(-j2\pi f_{N_k} t_1) \end{matrix} \right\|}} \right\} N_k \text{ řádků} \cdot \quad (5.41)$$

- 7) Výpočet matice \mathbf{M}_2 Dopplerově posunutých signálů, kde každý řádek tvoří přímý signál s jedním časovým posunem:

$$\mathbf{M}_2 = \mathbf{M}_{\text{pom2}} \cdot \mathbf{M}_{\text{Dop}} \quad (5.42)$$

Rozměr matice \mathbf{M}_2 je N_k řádků x N sloupců.

8) Matice **CAF** s N_τ řádky a N_k sloupce je dána vztahem:

$$\mathbf{CAF}(m, k) = \frac{1}{N} (\mathbf{M}_2 * \mathbf{M}_1)^T \quad (5.43)$$

Algoritmus maticové metody (MM):

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána.
2. Algoritmus vlastního výpočtu CAF
 - vytvoření matice časově posunutých referenčních signálů \mathbf{M}_1 odpovídající požadovanému rozsahu časových zpoždění (bistatických vzdáleností)
 - vytvoření matice frekvenčně posunutých referenčních signálů \mathbf{M}_2 odpovídající požadovanému rozsahu Dopplerových posunů
 - vlastní výpočet vzájemné funkce neurčitosti

$$\mathbf{CAF}(m, k) = \frac{1}{N} (\mathbf{M}_2 * \mathbf{M}_1)^T$$

3. Stanovení maxima CAF(m, k)
 - a. Řez v maximu funkce CAF(m, k_{max}) → interpolace pro získání vyššího rozlišení ve vzdálenosti v rámci buňky, ve které se nachází cíl → nalezení maxima $m_{max} = cm_{max}$
 - b. Řez v maximu funkce CAF(m_{max}, k) → interpolace pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl → nalezení maxima k_{max}

Tab. 5. Algoritmus výpočtu CAF maticovou metodou (MM)

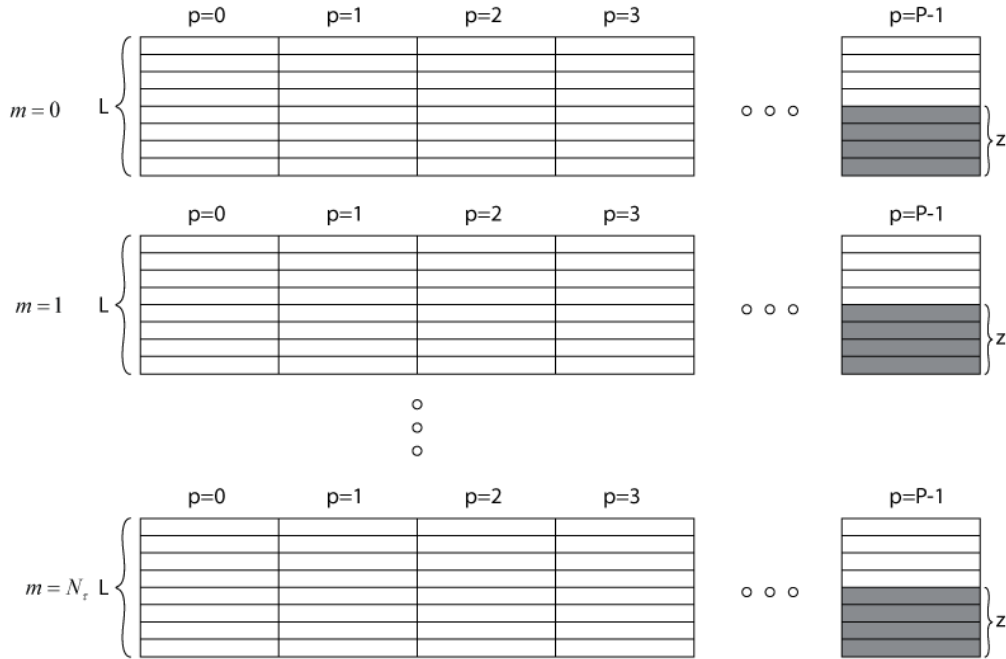
5.5.5 Metoda výpočtu CAF pomocí FFT s decimací signálového součinu (FFTD)

Výpočet pomocí metod bez rozdělení signálu do bloků může být vhodný pro specifické aplikace, ale pro většinu aplikací je tato metoda příliš náročná na výkon použitého hardwaru (zvláště při požadavku na zpracování v reálném čase). Následující metoda vychází z metody FFT definované vztahem (5.35). Na rozdíl od této metody však nedochází k výpočtu FFT ze signálového součinu na celém rozsahu dat N , ale pouze na bloku dat, který je určen decimačním poměrem. Díky decimaci signálového součinu dojde ke snížení počtu numerických operací a snížení délky výpočtu (viz Obr. 16), ovšem za cenu zhoršení přesnosti určení polohy vrcholu $|\mathbf{CAF}(m, k)|_{max}$.

Prvním krokem, je rozdělení signálu do P bloků čemuž bude odpovídat

$$P = \frac{N + z}{L}, \quad (5.44)$$

kde N je celkový počet vzorků, z je počet nulových vzorků dodaných do bloku, a to z důvodu potřeby stejné délky bloků z intervalu $l=0..L-1$ a L je označení počtu vzorků v každém bloku, tj. velikosti decimálního poměru. Grafické zobrazení rozdělení signálu do P bloků je zobrazeno na Obr. 15.



Obr. 15 Grafické zobrazení rozdělení vzorků do m skupin

Dosadíme-li vztah (5.44) do vztahu (5.35) dostaneme

$$CAF(m, k) = \sum_{l=0}^{L-1} \sum_{p=0}^{P-1} r_m[pL+l] \exp\left(\frac{-j2\pi k(pL+l)}{N}\right), \quad (5.45)$$

jehož úpravou získáme

$$CAF(m, k) = \sum_{p=0}^{P-1} \exp\left(\frac{-j2\pi kpL}{N}\right) \sum_{l=0}^{L-1} r_m[pL+l] \exp\left(\frac{-j2\pi kl}{N}\right). \quad (5.46)$$

Vztah (5.46) lze zjednodušit za předpokladu, že fázový posuv $\exp\left(\frac{-j2\pi kpL}{N}\right)$ v důsledku Dopplerova jevu bude v rámci jednoho bloku $l=0,1,2,\dots,L-1$ zanedbatelný. Protože

víme, že $F_{\max} = \frac{N_k}{T} \ll f_s \Rightarrow N_k \ll Tf_s = N$, bude nutné volit délku bloků L tak, že

exponent $\frac{2\pi lk}{N} < \frac{2\pi LN_k}{N} \ll 2\pi$. K tomu je nutné dodržet podmínku $L \ll \frac{N}{N_k} = \frac{f_s}{F_{\max}}$.

Tato úprava povede ke zjednodušení vztahu (5.46) na tvar

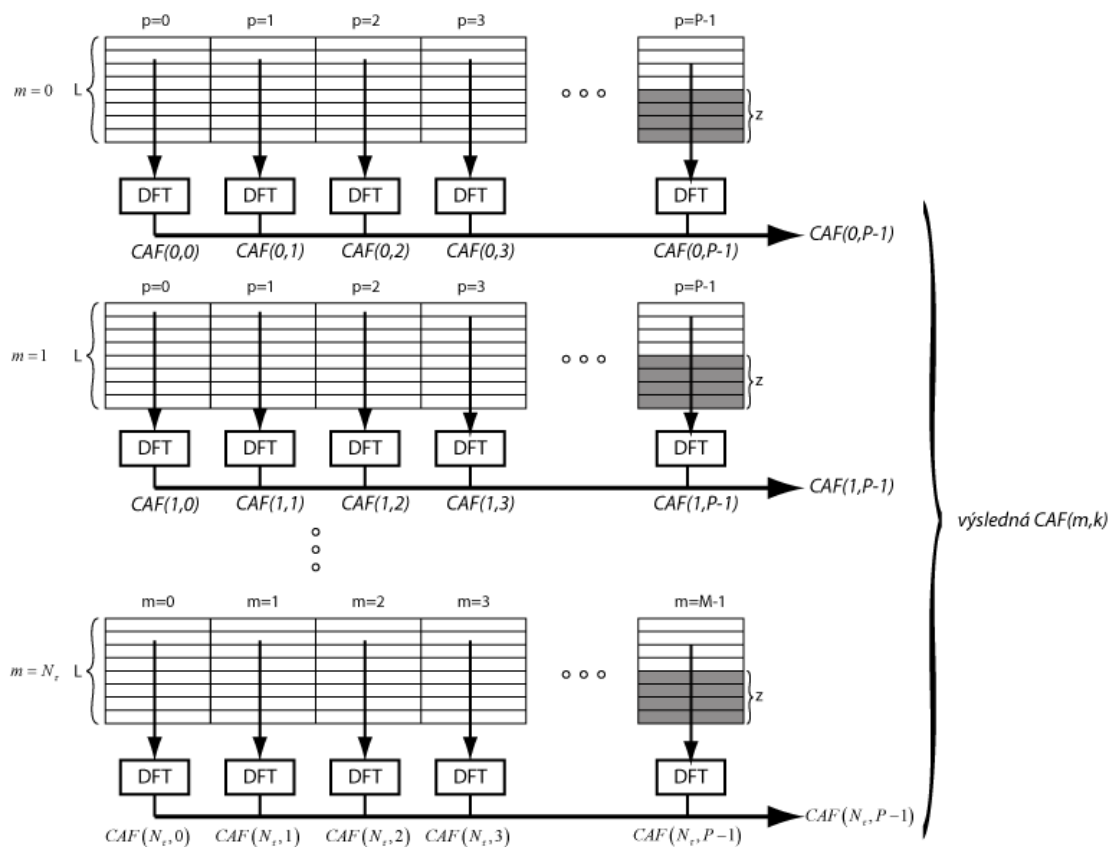
$$CAF(m, k) = \sum_{p=0}^{P-1} \exp\left(\frac{-j2\pi kpL}{N}\right) \sum_{l=0}^{L-1} r_m[pL+l]. \quad (5.47)$$

Vztah (5.47) je ekvivalentním výsledkem metody výpočtu CAF označené jako „Fine method“ v [40].

Označíme-li vnitřní sumu $\tilde{r}_m[p] = \sum_{l=0}^{L-1} r_m[pL+l]$, pak lze vztah (5.47) přepsat do tvaru

$$CAF(m, k) = \sum_{p=0}^{P-1} \tilde{r}_m[p] \exp\left(\frac{-j2\pi kpL}{N}\right). \quad (5.48)$$

Vztah (5.48) představuje konečný vztah pro výpočet vzájemné funkce neurčitosti s decimací signálového součinu. Grafické zobrazení postupu výpočtu $CAF(m, k)$ je zobrazeno na Obr. 16.



Obr. 16 Grafické zobrazení výpočtu CAF pomocí metody FFT s decimací signálového součinu

Algoritmus FFT metody s decimací signálového součinu (FFTD):

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána.
2. Stanovení velikosti decimace (filtrace pravoúhlým filtrem) délky L
3. Algoritmus vlastního výpočtu CAF
 - for $k = 0$ do $R_{bin} - 1$
 - výpočet signálového součinu přímého a posunutého signálu $r_m[n]$
 - for $p = 0$ do $P - 1$
 - decimace $r_m[n]$ decimačním faktorem L , výsledkem je $\tilde{r}_{mdec}[p]$
 - $$CAF_{full}(m, p) = \frac{1}{N} FFT \{ \tilde{r}_{mdec}[p] \}$$
 - end;
 - end;
5. Ořez funkce $CAF(m, p)$ vypočtené pro všechny $p \in \langle 0, P \rangle$ na požadovaný rozsah $CAF(m, k)$ pro $k \in \langle 0, F_{bin} \rangle$
4. Stanovení maxima $CAF(m, k)$
 - a. Řez v maximu funkce $CAF(m, k_{max}) \rightarrow$ interpolace pro získání vyššího rozlišení ve vzdálenosti v rámci buňky, ve které se nachází cíl \rightarrow nalezení maxima m_{max}
 - b. Řez v maximu funkce $CAF(m_{max}, k) \rightarrow$ interpolace pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima k_{max}

Tab. 6. Algoritmus výpočtu CAF pomocí metody FFTD

5.5.6 Metoda výpočtu CAF ve frekvenční oblasti (FMGFD)

Metoda výpočtu CAF ve frekvenční oblasti využívá rozdělení vzorků obou signálů do P bloků po L vzorcích, což lze vyjádřit vztahem

$$\begin{aligned}
 CAF(m, k) &= \sum_{p=0}^{P-1} \sum_{l=0}^{L-1} s_T^*[pL+l] s_R[pL+l+m] \cdot \exp\left(\frac{-j2\pi k(pL+l)}{N}\right) \\
 &= \sum_{p=0}^{P-1} \exp\left(\frac{-j2\pi pL}{N}\right) \sum_{l=0}^{L-1} s_T^*[pL+l] s_R[pL+l+m] \cdot \exp\left(\frac{-j2\pi kl}{N}\right)
 \end{aligned} \quad (5.49)$$

Vztah (5.49) lze zjednodušit podle stejných předpokladů, které jsou uvedeny v kapitole 5.5.5, viz zjednodušení ve vztahu (5.47), čímž vztah (5.49) přejde na tvar

$$CAF(m, k) = \sum_{p=0}^{P-1} \exp\left(\frac{-j2\pi pL}{N}\right) \sum_{l=0}^{L-1} s_T^*[pL+l] s_R[pL+l+m]. \quad (5.50)$$

V každém bloku jsou vzorky doplněny L nulovými vzorky, takže dostaneme P bloků po $2L$ vzorcích obou signálů

$$\begin{aligned} s_{T1}(2pL+l) &= s_T(pL+l) \text{ pro } l=0,1,\dots,L-1 \\ &= 0 \text{ pro ostatní } l \\ s_{R1}(2pL+l) &= s_R(pL+l) \text{ pro } l=0,1,\dots,L-1 \\ &= 0 \text{ pro ostatní } l \end{aligned} \quad (5.51)$$

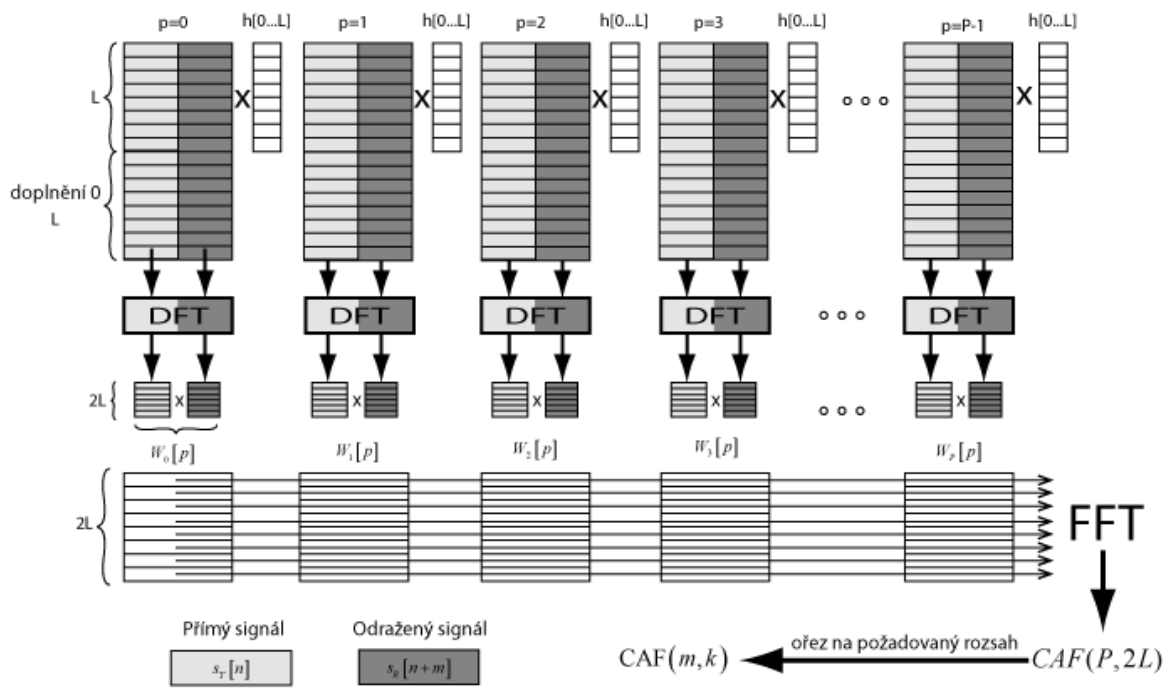
Následně se bloky transformují do frekvenční oblasti pomocí Fourierovy transformace, kde se vynásobí spektra signálů s_{T1} a s_{R1} v jednotlivých blocích a transformují zpět do časové oblasti

$$R_p[d] = F^{-1} \left[F[s_{T1}^*(pL+l)] \cdot F[s_{R1}(pL+l+m)] \right], \text{ pro } d=0,1,\dots,2L-1. \quad (5.52)$$

Výsledná CAF je vypočítána pomocí DFT přes všechny Dopplerovské frekvence.

$$CAF(d, p) = \sum_{p=0}^{P-1} R_p[d] \exp\left(\frac{-j2\pi pL}{N}\right), \quad (5.53)$$

což představuje výsledný vztah pro výpočet CAF ve frekvenční oblasti v [41]. Schéma algoritmu je zobrazeno na Obr. 17. Z důvodu porovnání výsledných povrchů CAF dojde k oříznutí vypočtené funkce na rozměry odpovídající $CAF(m, k)$.



Obr. 17 Grafické zobrazení výpočtu CAF pomocí metody ve frekvenční oblasti

Algoritmus metody výpočtu CAF ve frekvenční oblasti:

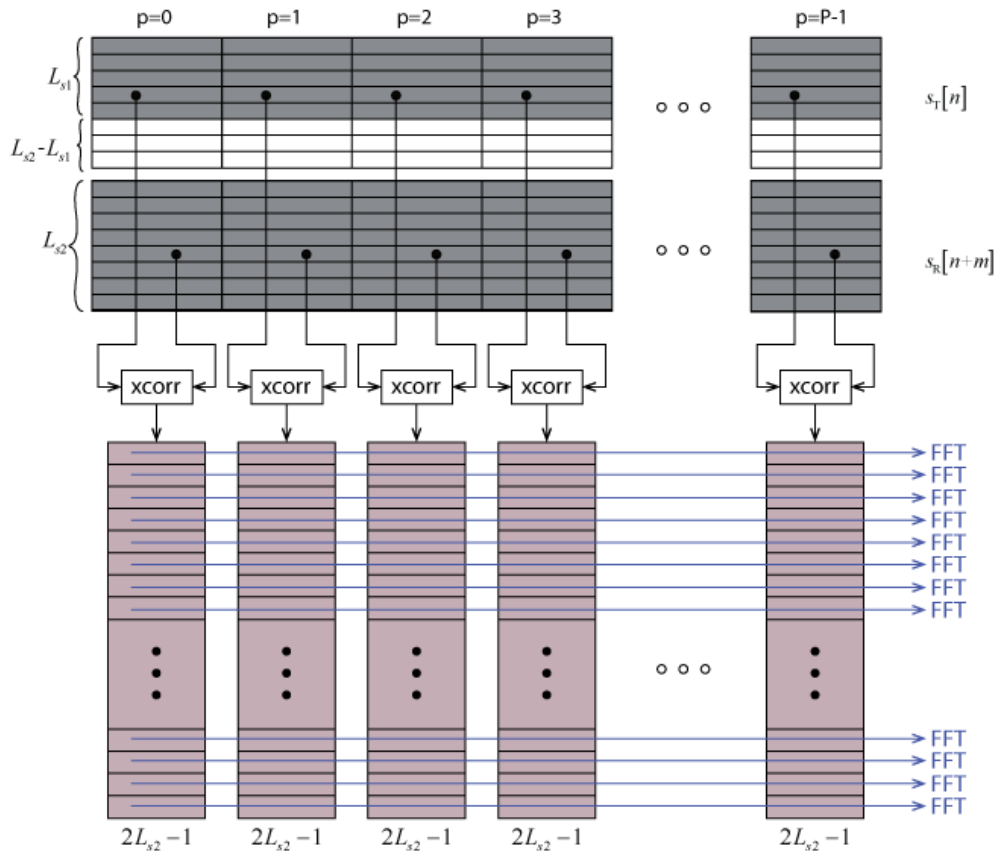
1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána
2. Stanovení velikosti bloku L
3. Algoritmus vlastního výpočtu CAF
 - for $p = 0$ do $P - 1$
 - for $n = 0$ do $L - 1$
 - $s_T^*[pL + l], s_R[pL + l + m]$
 - end
 - doplnění, každého bloku nulami do velikosti $d = 2L - 1$
 - $s_{T1}^*[2pL + l], s_{R1}[2pL + l + m]$
 - výpočet $R_p[d] = F^{-1} \left[F \left[s_{T1}^*(pL + l) \right] \cdot F \left[s_{R1}(pL + l + m) \right] \right]$
 - end;
4. Výpočet $CAF(d, p) = \sum_{p=0}^{P-1} R_p[d] \exp\left(\frac{-j2\pi kpL}{N}\right)$
5. Ořez funkce $CAF(d, p)$ na požadovaný rozsah $CAF(m, k)$
6. Stanovení maxima $CAF(m, k)$
 - a. Řez v maximu funkce $CAF(m, k_{max}) \rightarrow$ interpolace pro získání vyššího rozlišení ve vzdálenosti (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima m_{max}
 - b. Řez v maximu funkce $CAF(m_{max}, k) \rightarrow$ interpolace pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima k_{max}

Tab. 7. Algoritmus výpočtu CAF ve frekvenční oblasti (FMGFD)

5.5.7 Metoda výpočtu CAF pomocí částečných korelací (FMCW)

Poslední zde uvedená metoda [66], je „převzatá“ z metod zpracování signálu známých zejména u frekvenčně modulovaných radarů, které využívají, jak již z názvu vyplývá, pro detekci frekvenčně modulované spojité signály. Zkratka FMCW pochází z anglického názvu Frequency Modulated Continuous Wave. Princip je založen na výpočtu vzájemné korelace *nestejně* dlouhých částí přímého a odraženého signálu a následného stanovení Dopplerova posunu pro jednotlivé bistatické vzdálenosti N_τ . Na výstupní matici částečných korelací pak napříč aplikujeme Fourierovu transformaci, jejíž velikost je dána počtem bloků, do kterých je rozdělen signál. Počet

bloků je shodný s počtem stanovených Dopplerovských posunů vypočtené CAF. Grafické zobrazení algoritmu je zobrazené na Obr. 18.



Obr. 18 Princip metody výpočtu CAF založený na výpočtu částečných korelací

Pro stanovení jednotlivých parametrů vyjdeme ze znalosti potřebné velikosti vypočtené CAF, N_τ je počet binů v požadovaném rozsahu bistatických vzdáleností a N_k je počet binů v požadovaném rozsahu Dopplerova posunu.

Rozlišení CAF v bistatické vzdálenosti je dáno rozdílem počtu vzorků odraženého L_{s1} a přímého signálu L_{s2}

$$L_{s1} - L_{s2} \geq N_\tau. \quad (5.54)$$

Maximální vzdálenost je pak dána vztahem

$$R_{\max} = (L_{s1} - L_{s2}) dR = (L_{s1} - L_{s2}) \frac{c}{f_s} [m]. \quad (5.55)$$

Maximální Doppler je dán délkou referenčního signálu L_{s2} a je určen vztahem

$$F_{\max} = L_{s2} dF = L_{s2} \frac{f_s}{N}. \quad (5.56)$$

Délku odraženého signálu L_{s2} lze stanovit s využitím vztahu (5.54) a znalostí počtu binů ve vzdálenosti a Doppleru

$$L_{s1} \geq N_\tau + N_k. \quad (5.57)$$

Rozlišení *CAF* v Doppleru je svázáno s počtem bloků P_1 , do kterých je signál rozdělen, což lze vyjádřit vztahem

$$P_1 \geq N_k. \quad (5.58)$$

Provedeme-li rozdělení signálu do P_2 bloků, je jejich počet určen vztahem

$$P_2 = \{N / L_{s1}\}_{p=1}^{P_2}, \quad (5.59)$$

kde L_{s1} je délka bloku odraženého signálu, N celkový počet vzorků signálu.

Kombinací podmínek (5.57) - (5.59) se můžeme dostat do rozporu $P_1 \neq P_2$, kdy počet bloků stanovených v (5.59) není shodný s počtem bloků daných podmínkou (5.58) a tudíž nedojde ke splnění podmínky pro délku bloku odraženého signálu L_{s1} . Obecně lze tento rozpor vyjádřit pomocí vztahu $P \cdot L_{s1} > N$, kde P je počet bloků.

Řešením může být zvýšení počtu vzorků signálu, které je ale nevhodné, protože dochází ke snížení kroku ve frekvenční ose a tím ke snížení maximální určené Dopplerovy frekvence (5.56), či zvýšení počtu bloků, které je nutno překrývat přes sebe.

Překryv jednotlivých bloků pro získání odpovídajícího rozlišení v Dopplerově posunu určíme pomocí vztahu

$$\text{překryv} = \text{ceil} \left(\frac{P \cdot L_{s1} - N}{P - 1} \right). \quad (5.60)$$

Zvolíme-li např. $N_\tau = 400$, $P_1 = N_k = 600$, $N = 10^5$, dostaneme z podmínky (5.57) $L_{s1} \geq 1000$; provedeme-li výpočet počtu bloků pomocí vztahu (5.59), dostaneme $P_2 = N / L_{s1} = 100$. Z porovnání počtu bloků P_1 a P_2 je zřejmý rozpor. Překryv jednotlivých bloků by byl stanoven pomocí vztahu (5.60) a jeho hodnota je rovna 435 vzorků.

Algoritmus metody CAF pomocí částečných korelací (FMCW):

1. Stanovení počtu bistatických vzdáleností R_{bin} a Dopplerových posunů F_{bin} , ve kterých má být CAF vypočítána.
2. Rozdělení přímého $s_T[n]$ a odraženého $s_R[n+m]$ signálu do M bloků (při nedostatečném rozlišení je nutné jednotlivé bloky překrývat) – výstupem jsou matice $\mathbf{S}_1 = [L_{s_2} \times M]$ a $\mathbf{S}_2 = [L_{s_2} \times M]$.
3. Doplnění matice \mathbf{S}_1 nulami (v intervalu $\{\mathbf{S}_1\}_{L_{s_1}}^{L_{s_2}} = 0$, jejichž počet je určen vztahem $L_{s_2} - L_{s_1}$).
4. Výpočet částečných korelací přes jednotlivé bloky přímého a odraženého signálu
for $k = 0$ do M

$$Par_Cor(2L_{s_2}, M) = xcorr(\{\mathbf{S}_1\}_0^M, \{\mathbf{S}_2\}_0^M)$$
end;
5. Vlastní výpočet $CAF(L_{s_2} - L_{s_1} - 1, M) = FFT(Par_Cor(2L_{s_2}, M)^T)$
6. Ořez funkce $CAF(L_{s_2} - L_{s_1} - 1, M)$ na požadovaný rozsah $CAF(m, k)$
7. Stanovení maxima $CAF(m, k)$
 - a. Řez v maximu funkce $CAF(m, k_{max}) \rightarrow$ interpolace pro získání vyššího rozlišení ve vzdálenosti v rámci buňky, ve které se nachází cíl \rightarrow nalezení maxima m_{max}
 - b. Řez v maximu funkce $CAF(m_{max}, k) \rightarrow$ interpolace, pro získání vyššího rozlišení v Doppleru (v rámci buňky), ve které se nachází cíl \rightarrow nalezení maxima k_{max}

Tab. 8. Algoritmus výpočtu CAF pomocí metody částečných korelací (FMCW)

6 VÝPOČET FUNKCE NEURČITOSTI V REALTIME APLIKACÍCH

Doba výpočtu vzájemné funkce neurčitosti a přesnost určení maxima CAF je stěžejním parametrem určujícím jeho možnost nasazení v aplikacích zpracovávajících data v reálném čase, jako jsou radarové systémy PCL (viz kapitola 3.4). Zrychlení doby výpočtu lze provést na základě čtyř principálně odlišných přístupů nebo jejich kombinací:

- 1) Výpočet CAF založený na redukci vstupních dat (sériové zpracování dat);
- 2) Výpočet CAF s využitím vícejádrových CPU / víceprocesorových PC (paralelní zpracování dat);
- 3) Výpočet CAF s využitím clusteru (distribuované zpracování);
- 4) Výpočet CAF s využitím výkonu grafické karty, tzv. GPU computing (paralelní zpracování dat).

Každý přístup je spojen s jinými nároky na techniku programování, a to díky rozdílu mezi sériovým a paralelním zpracováním dat. Před podrobnějším vysvětlením jednotlivých přístupů pro zrychlení doby výpočtu CAF bude nejdříve nastíněn rozdíl a specifikace mezi sériovým a paralelním zpracováním dat.

Sériové (sekvenční) programování je založeno na řešení problému pomocí série po sobě jdoucích instrukcí, což znamená, že v procesoru PC se v jeden okamžik může zpracovávat pouze jedna instrukce. Jedná se o standardní programovací techniky, které jsou běžně využívány.

S rozvojem jedno-jádrových procesorů, založeným zejména na zvyšování frekvence CPU a zvyšování počtu tranzistorů, bylo dosaženo fyzických omezení (zejména odvod vyzařovaného tepla) zabraňujících dalšímu zvyšování frekvence CPU. Tento problém byl vyřešen pomocí vícejádrových, tzv. Multi-core procesorů, které obsahují více jader pracujících nezávisle na sobě a tím umožňují zpracovávat úlohu paralelně. Numerický výpočet pak lze zpracovávat paralelním způsobem, kdy každé jádro zpracovává určitý blok vstupních dat v jeden časový okamžik. Paralelnímu zpracování dat je věnováno velké množství literatury, např. [51]-[54].

Je nutno zmínit, že zdrojové kódy naprogramované pro sériové zpracování, nelze až na výjimky použít pro paralelní zpracování bez úpravy zdrojového kódu vhodného pro paralelní běh programu. Zásadním problémem u paralelního programování je synchronizace I/O operací, a to hlavně vzhledem k přístupu do paměti. Je třeba ošetřit situace, kdy dva či více procesů přistupují do jednoho místa v paměti, které se snaží přepsat. Stejně obtíže jsou i na úrovni I/O diskových operací. Specifika a odlišnosti paralelního programování jsou podrobně popsány v [53]-[55].

Transformace sériových algoritmů do paralelního provedení se označuje také slovem vektorizace. Vektorizace algoritmů je často velice náročný proces, jenž nemusí mít vždy řešení, vždy záleží na konkrétním algoritmu, který převádíme. Podrobné informace o vektorizaci dat lze nalézt v sérii publikací zabývajících se pouze tímto tématem [56]-[58]. Samostatnou kapitolou je paralelizace (vektorizace) cyklů, jejichž použití je v paralelním programování omezené ve srovnání s programováním sériových algoritmů.

Mezi nejvýznamnější odlišnosti patří u cyklů zejména:

1. Nemožnost paralelizovat vnořené cykly;
2. Nemožnost měnit proměnnou uvnitř cyklu (je určena pouze pro čtení);
3. Při zpracování cyklu není garantováno pořadí indexu smyčky;
4. Průchod cyklem nesmí záviset na předchozím průchodu cyklem.

Více podrobností o omezení použití cyklů lze najít v [57].

6.1 Výpočet CAF založený na dělení vstupních dat do bloků

První z principů je založen na výpočtu CAF po blocích vstupních dat, díky němuž dochází k nižší náročnosti na výpočet CAF. Nevýhodou tohoto přístupu je poškození fázové informace obsažené v signálu. Tento efekt se označuje jako dekoherence, která má zásadní vliv na přesnost určení maxima funkce CAF a následnému zhoršení odhadu pozice cíle. Zpracování algoritmů výpočtu CAF probíhá pomocí sériového zpracování kódu.

6.2 Výpočet CAF s využitím vícejádrových CPU, víceprocesorových PC, clusteru

Využití vícejádrových CPU, víceprocesorových PC či clusteru, je založeno na paralelním zpracování dat. Programování algoritmů pro tyto výpočetní systémy je shodné, proto jejich popis bude uveden ve společné kapitole s uvedením rozdílů jednotlivých systémů.

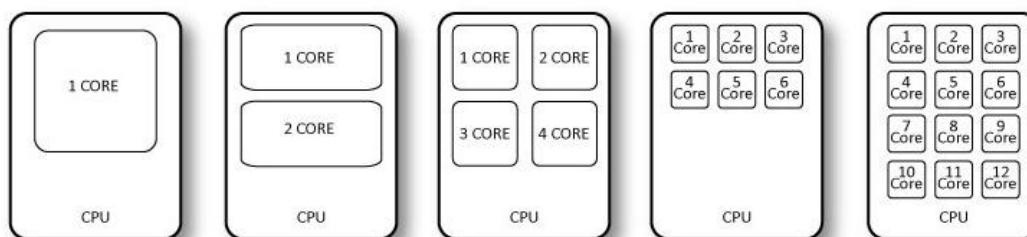
Z hlediska historického vývoje první byly výpočetní systémy založeny na jednojádrových procesorech, obsahujících, jak již z názvu vyplývá, jedno výpočetní jádro, které v jeden časový okamžik dokázalo zpracovat pouze jednu instrukci.

Dalším mezníkem byl vývoj vícejádrových procesorů, které již dokázaly s vhodným softwarovým vybavením zpracovávat data paralelně. Každé jádro v daném procesoru pracuje nezávisle na ostatních, čímž při zpracování dochází k úspoře výpočetní doby. Ukázka principiálního zapojení vícejádrových procesorů je na Obr. 19. Každý

blok představuje jeden procesor. Teoretická doba výpočtu u vícejádrového procesoru T_{multi} bude dána vztahem

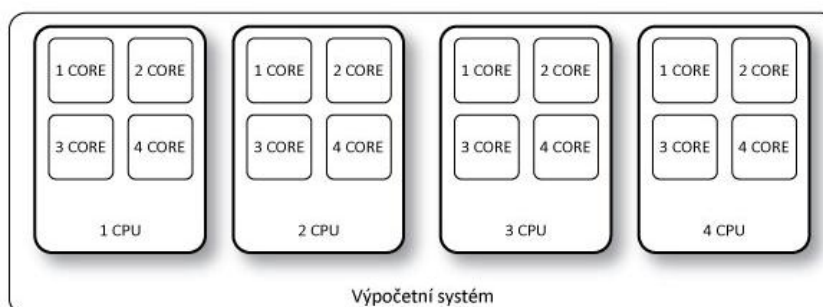
$$T_{multi} = \frac{T_{single}}{n}, \quad (6.1)$$

kde T_{single} je doba výpočtu jednojádrového procesoru [s] a n je počet jader v procesoru. Ve skutečnosti nedojde k tak velké časové úspoře, což je způsobeno „komunikací“ jader s rozhraním procesoru, skládání částečných výsledků z každého jádra do celku, atd.



Obr. 19 Principiální zapojení vícejádrových procesorů

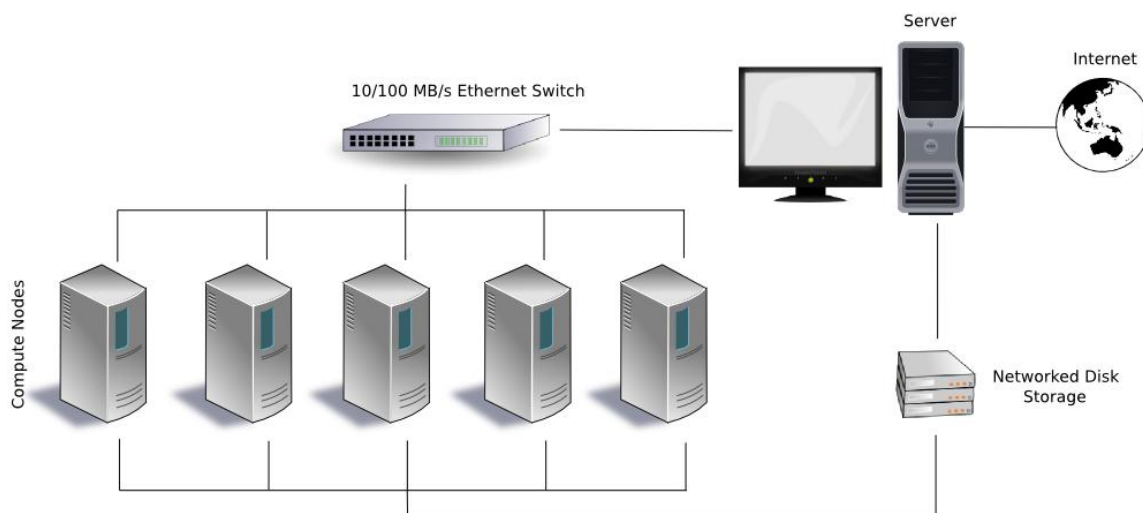
Víceprocesorový systém je PC stanice, která obsahuje dva či více procesorů. Toto uspořádání vzniklo z důvodu zvýšení výpočetního výkonu. Víceprocesorové systémy využívají sdílené paměti, což klade velké nároky na její velikost (v porovnání s jednoprocessorovými stanicemi). Každý z procesorů může být tvořen jedno či více jádrovým procesorem. Principiální zapojení 4 procesorového systému, kde každý z procesorů obsahuje 4 výpočetní jádra, je zobrazeno na Obr. 20. Vztah (6.1) lze použít i pro víceprocesorové systémy, kde za n dosadíme celkový počet jader všech procesorů.



Obr. 20 Principiální zapojení víceprocesorových systémů

Rozvoj vysokorychlostních sítí otevřel možnost spojování jednotlivých výpočetních systémů do jednoho celku, tzv. *clusteru*. Cluster je skupina volně vázaných počítačů, serverů a jiných výpočetních komponent, které spolu spolupracují na konkrétních numerických úlohách a z pohledu jejich zpracování pracují jako jeden výpočetní

system. Myšlenka využití více počítačů spojených v jeden systém je poměrně stará, první zmínka o využití je již z roku 1967. Podrobné informace o jejich vývoji lze nalézt v [47]-[50]. Podle zaměření rozdělujeme clustery do několika skupin, přičemž pro výpočetní (numerické) úlohy, jsou používány tzv. výpočetní clustery. Výpočetní clustery jsou často označovány zkratkou HPC clustery, pocházející z anglického spojení slov High-Performance Computing. Ukázka principiálního zapojení clusteru je na Obr. 21.



Obr. 21 Principiální zapojení clusteru [49]

Pro pochopení principu práce výpočetního clusteru je nutné znát pojmy, které jsou s tímto tématem svázané. Cluster se skládá z jednotlivých výpočetních PC stanic, tzv. výpočetních uzlů (název pochází z anglické terminologie *nodes*). Jednotlivé uzly jsou propojeny vysokorychlostní sítí, která je připojena na server, tzv. head node (hlavní výpočetní uzel). Head node může tvořit nejen nadřazený server, ale může být tvořen i „normálním“ uzlem (jedním z výpočetních uzlů). Jednotlivé výpočetní uzly mohou být tvořeny různorodými (výše uvedenými) výpočetními systémy. Z pohledu uživatele zpracovávajícího numerickou úlohu se cluster jeví jako jedno PC.

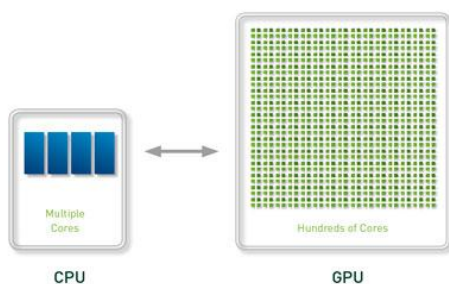
Numerický výpočet úloh na clusteru lze rozdělit podle komunikace (přenosu dat) mezi jednotlivými uzly clusteru do dvou oblastí:

1. Head node komunikuje se všemi výpočetními uzly, ale jednotlivé výpočetní uzly nekomunikují mezi sebou (nedochází k žádné výměně dat). Tento případ spadá do kategorie distribuovaného zpracování. Výpočet *CAF* spadá do této kategorie.
2. Head node a i jednotlivé výpočetní uzly komunikují při výpočtu mezi sebou. Tento druhý případ je podstatně náročnější na ošetření všech podmínek, aby

nedocházelo k přepisu I/O dat v paměti jednotlivých nodů či ke stejnému problému u diskových operací.

6.3 Výpočet CAF s využitím grafické karty GPU

Požadavky uživatelů na stále vyšší výpočetní výkon za „rozumnou cenu“ vedly k myšlence využití grafických karet pro numerické výpočty. Jedná se o historicky nejmladší možnost zvýšení výpočetního výkonu, která se označuje jako výpočty pomocí GPU, kde GPU označuje zkratku slov Graphic Processor Unit – grafická procesorová jednotka. Finanční náklady výpočetních systémů, které jsou založené na víceprocesorových stanicích, clusterech, atd. jsou vysoké a tím často i neefektivní. Jestliže tyto finanční náklady porovnáme s výpočetními systémy využívající GPU, pak systém GPU představuje významnou finanční úsporu při stejném výpočetním výkonu. Výhoda využití GPU je zejména v tom, že grafické zobrazení je založeno na matematických výpočtech (vektorový, maticový počet), které je blízké numerickým řešením úloh v mnoha oblastech lidské činnosti. Důležitým hlediskem CPU a GPU systémů je porovnání počtu jader. Běžně dostupné CPU procesory obsahují cca 12 jader, zatímco počet jader GPU je v řádu stovek, viz Obr. 22.



Obr. 22 Porovnání CPU a GPU z hlediska počtu jader [59]

Z využití velkého množství jader v GPU plyne zřejmá výhoda využití těchto karet pro numerické úlohy, kde každé jádro GPU může *samostatně* zpracovávat určitou část numerického výpočtu. Problematika využití GPU je velice obsáhlá, rychle se rozvíjející, z tohoto důvodu uvádím odkazy na literaturu [59]-[61].

Programování pro GPU má svá specifika, odlišná od sériového programování či programování pro cluster, a to zejména z důvodu množství využitelné operační paměti RAM výpočetního systému. V běžných výpočetních systémech (stolní PC, clustery, víceprocesorové stanice) není velikost paměti RAM nijak zvlášť omezujícím faktorem. Běžně dostupné výpočetní systémy mohou mít operační paměť RAM v řádu jednotek až desítek GB. Rozšíření operační paměti RAM není z pohledu

finančních nákladů nijak omezující a je technicky běžně dostupné. Numerické řešení výpočetně a paměťově náročných úloh tak nepředstavuje žádné komplikace.

U programovacích technik využívajících grafické karty je velikost paměti v řádu jednotek GB. V současné době nejvýkonnější grafické karty od firmy Nvidia - Tesla C2075 a Quadro 6000 mají 6 GB paměti, což vyžaduje ve většině případů naprosto jiný přístup ve stylu programování pro CPU a GPU. Programovací techniky pro GPU jsou založeny na rychlých výpočtech malých bloků dat, jež jsou pak „převedeny“ zpět do CPU (respektive operační paměti RAM). Důležitým aspektem GPU programování je minimalizace komunikace mezi CPU a GPU, která jinak vede k podstatnému prodloužení doby výpočtu. Obecně lze styl GPU programování shrnout do několika níže uvedených kroků:

- 1) Rozdělení vstupních dat do menších bloků, kde velikost bloku bude záležet na počtu a náročnosti matematických operací z hlediska velikosti paměti grafické karty.
- 2) Převod bloku dat do GPU ($CPU \rightarrow GPU$).
- 3) Provedení všech potřebných matematických operací s odpovídajícím blokem dat v GPU (paralelní zpracování). Je nutné zabránit transferu částečných mezivýpočtů do CPU a jejich následného převodu zpět do GPU. Z důvodu co nejvyšší optimalizace je nutno provádět veškeré numerické operace přímo v GPU.
- 4) Převod výsledného bloku dat do CPU ($GPU \rightarrow CPU$).
- 5) Operace uvedené v bodech 1-4 se provedou pro každý blok dat.

Je nutno poznamenat, že v případě, kdy numerická úloha bude náročná „pouze“ z hlediska výpočetního výkonu a ne z hlediska náročnosti na paměť, pak nemusejí být vstupní data rozdělena do jednotlivých bloků, ale výpočet se provádí najednou.

6.4 Softwarové a HW vybavení pro paralelní programování

V této kapitole je popsáno softwarové a hardwarové vybavení, které bylo použito pro paralelní programování v této práci.

6.4.1 Softwarové vybavení

Operační systém je tvořen 64 bit verzí Microsoft Windows 7 Enterprise CZ, SP1. Jako výpočetní matematický software byl použit produkt od firmy **Mathworks MATLAB™** [50], který představuje kompletní výpočetní systém pro vědeckotechnické výpočty, modelování, návrhy algoritmů, paralelní výpočty, atd. Tento systém byl doplněn o následující rozšíření:

MATLAB Parallel Computing Toolbox™ [62] - rozšíření pro využití paralelních výpočtů v prostředí Matlab až pro 8 výpočetních jader (v rámci jednoho PC).

MATLAB Distributed Computing Server™ [63] - rozšíření pro využití clusterových paralelních výpočtů (komunikace více PC dohromady).

AccelerEyes Jacket® [64] – rozšíření Matlabu od firmy AccelerEyes, které slouží ke GPU výpočtům.

6.4.2 Hardwarové vybavení

Hardwarové vybavení lze rozdělit do dvou částí. První část tvoří HW vybavení clusteru, druhou část tvoří HW vybavení PC pro numerické výpočty využívající grafickou kartu GPU.

Specifikace HW clusteru:

Výpočetní cluster obsahuje 6 identických výpočetních uzlů obsahujících PC stanice

Dell Optiplex 790 složené z následujících (vybraných) komponent:

- Procesor: Intel® Core™ i5-2400 (6MB cache, 3.10 GHz)
- Operační paměť RAM: 12GB (1333MHZ DDR3 ECC)
- Pevný disk: Seagate, 320GB (SERIAL ATA II)
- Síťová karta: 1GB/s

HW výpočetní stanice pro numerické výpočty pomocí GPU je tvořen PC stanicí Dell Precision Workstation T3500 obsahující:

- Procesor: Intel® Xeon® Processor W3530 (8MB cache, 2.80 GHz)
- Operační paměť RAM: 4 GB (1333MHZ DDR3 ECC)
- Pevný disk: Seagate Barracuda 7200.12, 1 TB (SERIAL ATA II)
- Grafická karta: Gigabyte Nvidia GeForce GTX 580, 3072 MB GDDR5, PCI-E, obchodní označení GV-N580UD-3GI

7 MĚŘENÍ A ANALÝZA NUMERICKÉ NÁROČNOSTI CAF

Tato kapitola je věnována měření a analýze numerické náročnosti výpočtu *CAF* a je členěna do několika navazujících podkapitol. V první části dojde k popisu metodologie testování numerické náročnosti, poté k popisu vstupních parametrů vycházejících z reálných systémů. Následovat bude podkapitola věnující se vlastnímu měření numerické náročnosti výpočtu *CAF*, tj. stanovení doby výpočtu u jednotlivých metod výpočtu vzájemné funkce neurčitosti, s diskuzí naměřených výsledků. Následuje uvedení rozdílových ploch *CAF* mezi jednotlivými metodami pro účely určení vizuální „shody“ výsledné funkce neurčitosti. Závěr kapitoly je věnován celkovému zhodnocení a shrnutí výsledků.

7.1 Metodologie testování

Stanovení numerické náročnosti je v této práci provedeno měřením doby výpočtu pro jednotlivé, dále uvedené, testovací implementace metod výpočtu *CAF*. Tato metoda testování byla zvolena z důvodu praktických aplikací a závislosti realizace FLOP na konkrétním HW řešení. Důvody, které vedly k měření času doby výpočtu, jsou tyto:

- Doba výpočtu vs. počet numerických operací u moderních procesorů již nepředstavuje lineární závislost (např. zvýšení počtu numerických operací na dvojnásobek nepovede k dvojnásobnému zvýšení výpočetní doby) díky optimalizačním a akceleračním aritmetickým algoritmům implementovaným přímo v CPU resp. GPU.
- Doba vlastního výpočtu je (pro uživatele z praxe) směrodatnější a potřebnější hodnota (u reálných systémů je vždy stanovena maximální doba výpočtu, do které se vývojář tohoto systému musí „vejít“)
- Stanovení celkového počtu operací je u paralelního zpracování problematické, a to z důvodu *neznalosti* počtu režijních operací, tj. operací, které vznikají u
 - Clusteru - vlivem komunikace mezi řídicím uzlem a ostatními výpočetními uzly;
 - Nárůstu režijní komunikace s růstem počtu výpočetních uzlů;
 - Neznalosti počtu operací při inicializaci GPU karty pro numerický výpočet (a tím celkové doby potřebné pro inicializaci);
 - Neznalosti počtu operací mezi jednotlivými jádry GPU.

V odborné literatuře jsou tyto režijní operace označovány často používaným výrazem „overhead“ [65]. V této práci bude používán český ekvivalent výrazu, a to režijní operace či „režie“.

Je zřejmé, že měření doby výpočtu je nutné opakovat v dostatečném počtu a za předem stanovených podmínek, a to v důsledku získání co „nejvyšší“ objektivitu a relevantnosti výsledků založené na alespoň elementárním statistickém vyhodnocení. Všechna měření, která jsou součástí této práce, probíhala za shodných podmínek:

- Všechny výpočetní uzly (řídící uzel nevyjímaje) mají shodnou hardwarovou a softwarovou konfiguraci, tzv. homogenní cluster.
- V době měření výpočetní doby je spuštěn pouze SW Matlab a standardní služby operačního systému.
- Měření doby výpočtu jednotlivých implementací se provádělo opakovaně (20x), výsledná doba výpočtu pak byla stanovena aritmetickým průměrem těchto dvaceti měření.

Referenční metodou pro porovnání jednotlivých metod výpočtu *CAF* byla zvolena sumační metoda výpočtu v HW implementaci, kdy je pro výpočet využíváno pouze jedno jádro CPU (HW I.). Důvodem této volby byl fakt, že odpovídá definici vzájemné funkce neurčitosti. Je nutné podotknout, že sumační metoda dle definice obsahuje dva vnořené cykly, jejichž výpočet je velice neefektivní a řádově pomalejší než ostatní metody výpočtu *CAF*. Z tohoto důvodu došlo k „rozdělení“ dvou vnořených cyklů do dvou samostatných cyklů a tato metoda byla brána jako referenční (metoda 1 v Tab. 10).

Pro přehlednější vyhodnocení nejsou ve výsledcích uvedeny pouze doby výpočtu, ale i zrychlení (speedup) doby výpočtu, které je dáno vztahem

$$\text{Zrychlení} = \frac{CAF_sum}{CAF_x} [-], \quad (7.1)$$

kde *CAF_sum* představuje referenční dobu výpočtu *CAF* (konfigurace 1 v Tab. 9) a *CAF_x* představuje dobu výpočtu pro jednotlivé metody *CAF* pro různé HW konfigurace uvedené v Tab. 9.

Jednotlivé HW konfigurace lze rozdělit do 4 základních skupin (vyhodnocení z hlediska jednotlivých skupin je uvedeno v podkapitole 7.5.3):

- Sériové zpracování dat (výpočet probíhá na 1 jádru CPU);
- Paralelní zpracování – Multi-Core (výpočet probíhá na jedné PC stanici, ale s proměnným počtem jader na CPU umístěných na jedné základní desce);

- Paralelní zpracování – Cluster (výpočet probíhá na více PC stanicích s proměnným počtem jader na jednotlivých výpočetních uzlech propojených navzájem ethernet sítí);
- Paralelní zpracování – GPU (tvořené kombinací CPU a GPU, kde hlavní část výpočtu probíhá na GPU).

Sériové zpracování dat (referenční výpočet)			
Označení HW konfigurace	Počet výpočetních uzlů	Počet jader v jednotlivých výpočetních uzlech	Celkový počet jader
I.	1	1	1
Paralelní zpracování dat (Multi-Core) – 1 PC			
Označení HW konfigurace	Počet výpočetních uzlů	Počet jader v jednotlivých výpočetních uzlech	Celkový počet jader
II.	1	2	2
III.	1	4	4
Paralelní zpracování dat (Cluster) – více PC			
Označení HW konfigurace	Počet výpočetních uzlů	Počet jader v jednotlivých výpočetních uzlech	Celkový počet jader
IV.	2	1	2
V.	2	2	4
VI.	2	4	8
VII.	4	1	4
VIII.	4	2	8
IX.	4	4	16
Paralelní zpracování dat (grafická karta – GPU)			
X.	512 jader*		

Tab. 9. Seznam HW konfigurací

* Počet jader u GPU výpočtů nelze měnit, karta využívá pro výpočet vždy maximální počet dostupných jader.

Výsledky měření doby budou pro každou metodu shrnuty v tabulce, která bude obsahovat měření pro všechny HW konfigurace s uvedením průměrné doby výpočtu (aritmetický průměr).

Porovnání jednotlivých časů výpočtu algoritmů bude zobrazeno ve srovnávacích tabulkách pro jednotlivé HW konfigurace, kde pro každou konfiguraci dojde ke zvýraznění nejkratší a nejdelší doby výpočtu a uvedení faktoru zrychlení výpočtu. Celkové vyhodnocení je uvedeno v kapitole Tab. 28.

Součástí vyhodnocení bude i porovnání vypočtených ploch *CAF* pro jednotlivé algoritmy a výpočet jejich „rozdílu“ oproti referenční metodě. Výsledkem tohoto srovnání bude zobrazení rozdílové plochy *CAF*, jež umožní ukázat kvalitativní rozdíly mezi jednotlivými metodami.

Seznam testovaných metod (implementací) *CAF* je uveden v Tab. 10.

Algoritmus výpočtu CAF dle definice	
Název metody	Zkratka metody
Sumační metoda dle definice CAF (dva vnořené cykly)	SMD

Seznam testovaných metod výpočtu CAF		
Číslo metody	Název metody	Zkratka metody
1	Sumační metoda I. (referenční metoda)	SMI
2	Sumační metoda II.	SMII
3	Korelační metoda	XC
4	FFT metoda	FFT
5	FFT metoda s decimací signálového součinu I.	FFTD
6	Metoda výpočtu CAF ve frekvenční oblasti	FMGFD
7	Metoda výpočtu CAF pomocí částečných korelací	FMCW
8	Maticová metoda	MM

Tab. 10. Seznam testovaných metod výpočtu CAF

7.2 Vstupní parametry

V této podkapitole je uveden popis a zdůvodnění volby vstupních parametrů (využívaných dále pro vlastní výpočet CAF), které jsou dány typickými požadavky na reálný PCL systém. Při volbě vstupních parametrů jsem vycházel ze zveřejněných údajů experimentálních radarových systémů pracujících na tomto principu a také ze snahy o co největší obecnost při volbě vstupních parametrů, aby bylo možné jejich jednoduchou úpravou porovnávat zkoumané metody pro různé scénáře, které se mohou v praxi vyskytovat. Základními parametry PCL systémů jsou dosah radaru R_{\max} , jehož hodnota byla stanovena na úrovni 200 km. Maximální Dopplerův posun byl stanoven s ohledem na maximální rychlost cíle 700 ms^{-1} na úroveň $\pm 250 \text{ Hz}$.

Další parametry, jako je rozlišení radaru ve vzdálenosti, rozlišení radaru v Dopplerově posunu, počet binů ve vzdálenosti a Dopplerově posunu, jsou stanoveny pomocí vstupních parametrů, jako je vzorkovací frekvence, doba zpracovávaných signálů, atd. Vzorkovací frekvence použitá pro vzorkování vstupních signálů je $f_s = 200 \text{ kHz}$ a doba signálu zajišťující dostatečné rozlišení v bistatické vzdálenosti je stanovena na hodnotu $T = 1 \text{ s}$, počet vzorků zpracovávaného signálu je pak definován hodnotou $N = 2 \cdot 10^5$.

Základní rozlišení v Dopplerově posunu definované vztahem (5.23) je pak pro výše uvedené hodnoty $dF = 1 \text{ Hz}$. Základní rozlišení v bistatické vzdálenosti definované vztahem (5.22) s využitím znalosti, že rychlost šíření signálu je $c = 299792458 \text{ ms}^{-1}$, je $dR = 1498,96 \text{ m}$. Pro vyšší přehlednost jsou základní parametry, které jsou použité pro **všechna** měření v této práci, shrnuta v Tab. 11.

Vstupní signály lze generovat pomocí skriptu GenSigRand.m (umístěn na příloženém DVD), který využívá generátor pseudonáhodných signálů, či lze načíst libovolná naměřená data přímo v jednotlivých metodách výpočtu CAF. Vstupní signály použité v této práci jsou tvořeny pouze náhodnými signály, a to z důvodu jednoduchosti použití a také z důvodu zaměření této práce. Volbu polohy cíle lze ve skriptu měnit a to nastavením parametrů R a F, které určují polohu cíle ve vzdálenosti a Dopplerově posunu. Pro analýzu vypočtené CAF je ve skriptu definováno i nastavení odstavu signálu od šumu, a to zadáním oddělených hodnot pro přímý a odražený signál.

Volba vstupních dat není z hlediska numerické náročnosti výpočtu relevantní. Algoritmus zpracuje libovolná data, ač mohou být tvořena přímým a odraženým signálem získaným ze systémů FM, DVB-T, GSM atd.

Součástí každého algoritmu CAF je detekce vrcholu CAF odpovídajícího cíli, kde jeho souřadnice určují bistatickou vzdálenost a radiální rychlost cíle, která je svázaná se sítí uzlových bodů představujících v ose x jednotlivé bistatické vzdálenosti,

ve kterých se může cíl vyskytovat (interval hodnot $\langle 0; R_{\max} \rangle$) s rozlišením dR a osou y představující osu Dopplerovských posunů (interval hodnot $\langle -F_{\max}; F_{\max} \rangle$) s rozlišením dF . Z důvodu získání vyššího rozlišení v rámci jedné buňky byla zavedena interpolace, jež buňku (ať již v ose bistatické vzdálenosti či Dopplerova posunu) rozděluje na 10 dílčích částí. Teoreticky lze dosáhnout až rozlišení cca 150 m bistatické vzdálenosti a 0,1 Hz v Dopplerově posunu. Interpolace je započítávána do doby výpočtu každého algoritmu.

Vstupní parametry radarového systému (RS)	
Vlastnost RS	Hodnota
Délka vstupních signálů	1 [s]
Vzorkovací frekvence f_s	200 [kHz]
Počet vzorků přímého a odraženého signálu	200.000 [-]
Maximální dosah radaru v bistatické vzdálenosti R_{\max}	200 [km]
Maximální dosah radaru v Dopplerově posunu F_{\max}	± 250 [Hz]
Rozlišení v bistatické vzdálenosti dR	1498,96 [m]
Rozlišení v Dopplerově posunu dF	1 [Hz]

Tab. 11. Seznam vstupních parametrů

7.3 Měření dob výpočtu u jednotlivých metod CAF

Tato podkapitola je věnována zobrazení a diskuzi naměřených výsledků pro jednotlivé implementace výpočtu *CAF* provedených pro výše uvedené HW konfigurace. Součástí této podkapitoly je i popis odlišností a nároků jednotlivých algoritmů, které jsou pro každou implementaci jiné. Všechny časy uvedené v tabulkách jsou uvedeny v sekundách.

7.3.1 Sumační metoda dle definice (SMD)

Jedná se o základní metodu výpočtu *CAF*, jejíž výpočet je stanoven na základě definice vzájemné funkce neurčitosti. Tuto metodu lze najít na přiloženém DVD, pod jménem *A_CAF_sumace_DEF.m*. Důvodem umístění této metody do seznamu je ukázka důležitosti optimalizace, vektorizace a paralelizace algoritmů. Průměrná doba sériového výpočtu na jednom jádře je cca **18 min**. Tento extrémní čas je dán použitím dvou vnořených cyklů, které nelze žádným způsobem zpracovat v této

formě „paralelním“ způsobem, a také faktem, že systém Matlab neumí pracovat s vektory a maticemi v cyklech přes jejich indexy efektivně na rozdíl od low-level jazyků, jako jsou C/C++ nebo Fortran. Jako jediná z metod nemá uvedenou tabulku výsledků pro jednotlivé HW konfigurace, protože byla spočítána pouze pomocí sériového zpracování dat.

7.3.2 *Sumační metoda I.*

Sumační metoda I. je založena na využití algoritmu definovaného v kapitole 5.5.1. Algoritmus této metody je částečně optimalizován pro paralelní výpočet, a to rozdělením výpočtu do dvou nezávislých *for-end* cyklů, kde jednotlivé cykly využívají pro výpočet paralelní verzi výpočtu (využití cyklu *parfor-end*). Pro použití paralelní verze výpočtu obou cyklů bylo nutné upravit kód tak, aby jednotlivé iterace byly na sobě nezávislé z hlediska průchodu jednotlivými iteracemi. Naměřené hodnoty pro tuto metodu jsou uvedeny v Tab. 12. Na konci tabulky jsou uvedeny střední hodnoty naměřených časů pro jednotlivé HW konfigurace.

Metoda SMI										
HW konfigurace ----- Číslo měření	Sériové zpracování		Paralelní zpracování							GPU zpracování
	I.	Sériové zpracování	Multi-Core (1 PC stanice)		Cluster (více PC stanic)					
			II.	III.	IV.	V.	VI.	VII.	VIII.	
1	33,4614	18,8177	14,0663	21,8909	17,1510	11,3101	18,4532	18,4102	17,6384	0,8794
2	32,9831	17,9724	13,8389	23,0959	20,0725	14,7012	17,3904	18,3710	17,0445	0,8867
3	33,1242	17,8493	17,2414	22,1797	22,7349	14,0206	18,0671	17,7513	16,9524	0,8849
4	33,2286	17,8921	13,8685	27,2520	22,5661	11,5991	21,0941	16,9747	16,5408	0,8830
5	33,1819	18,1488	14,0826	24,1260	21,1054	11,2195	21,8336	17,3173	17,0305	0,8869
6	33,1622	18,1068	13,8336	25,2823	18,8491	14,3622	21,4233	16,9913	17,0399	0,8852
7	32,8997	18,1576	13,7011	25,5278	18,0518	13,3596	19,3138	19,7119	17,1987	0,8862
8	33,1460	17,9287	13,8389	21,1392	18,6581	11,4579	22,5872	17,1515	16,9787	0,8847
9	33,2384	18,2165	13,9715	22,0511	16,1041	14,3721	23,1339	17,3732	17,2776	0,8864
10	33,1823	18,0019	13,9217	24,4752	22,5945	14,1721	23,0058	17,7637	17,4954	0,8854
11	33,1924	18,2239	13,7891	25,8921	18,2882	11,4457	19,4964	18,0314	16,8855	0,8824
12	33,0726	18,1838	13,9608	27,1313	19,0795	11,5927	18,2234	16,9121	17,0568	0,8864
13	32,9600	21,0640	13,6617	28,0525	18,9045	11,2838	17,9420	16,2271	16,0506	0,8838
14	34,3012	18,2400	13,9679	24,8470	23,1725	13,2583	21,1620	18,5255	17,9000	0,8792
15	33,0891	20,7688	13,6772	27,4740	16,2097	14,5170	20,3266	18,2346	16,8023	0,8834
16	33,2191	18,0211	13,9779	25,2965	16,6866	12,0503	17,9217	21,8630	17,0402	0,8862
17	33,2257	18,2305	16,9693	22,3471	18,0610	11,0649	17,9861	16,9706	16,9872	0,8839
18	33,1610	21,1538	13,9071	21,1805	15,4001	12,0325	20,6792	16,2470	17,0800	0,8885
19	33,2512	21,4658	13,8797	26,5987	22,7334	11,0010	19,4501	21,2634	16,9644	0,8854
20	33,2513	18,1585	13,7321	24,9550	22,4938	14,1320	18,8868	22,1610	17,4098	0,8844
Aritmetický průměr	33,2166	18,7301	14,1944	24,5397	19,4458	12,6476	19,9188	18,2126	17,0687	0,8847
Minimální doba výpočtu	32,8997	17,8493	13,6617	21,1392	15,4001	11,0010	17,3904	16,2271	16,0506	0,8792
Maximální doba výpočtu	34,3012	21,4658	17,2414	28,0525	23,1725	14,7012	23,1339	22,1610	17,9000	0,8885

Tab. 12. Naměřené doby výpočtu CAF pomocí sumační metody I.

Z důvodu větší přehlednosti a názornějšího vyhodnocení jsou v Tab. 13 uvedeny průměrné doby výpočtu pro jednotlivé HW konfigurace s uvedením faktoru zrychlení doby výpočtu, jež je definován vztahem (7.1). Zvýrazněná hodnota představuje referenční čas, s kterým jsou porovnávány ostatní hodnoty (zvýraznění bude představovat referenční čas v souhrnných tabulkách u všech uvedených metod).

Z výsledných hodnot je patrné, že urychlení výpočtu v případě použití Multi-Core řešení (HW konfigurace II.-III.) ve srovnání se sériovým zpracováním (HW konfigurace I.) stoupá dle předpokladů vycházejících z lineárního speedup škálování. Teoretické zrychlení by mělo dosáhnout hodnoty 4 (díky počtu jader, které provádějí výpočet). Rozdíl je způsoben režií, která je způsobena komunikací mezi jádry a přesunem jednotlivých bloků dat mezi jádry a ostatními částmi systému.

Zajímavé srovnání lze také provést z hlediska porovnání výsledných hodnot mezi metodami II. a IV., které jsou srovnatelné co do počtu jader využitých pro výpočet CAF, ale rozdílné z hlediska počtu výpočetních stanic. V HW variantě II. jsou použita pro výpočet dvě jádra v rámci jednoho PC a ve variantě IV. je použito vždy jedno jádro na dvou výpočetních uzlech. Urychlení výpočtu kleslo z hodnoty 1,7734 na hodnotu 1,3536. Porovnáním těchto výsledků lze ukázat, že díky přesunu velkého množství dat po síti a komunikaci mezi jednotlivými výpočetními uzly klesá efektivnost výpočtu. Tento trend lze ukázat i na ostatních metodách o **shodném počtu jader**, ale rozdílném počtu použitých nodů. Jedná se o porovnání metod III. (nebo) V. vs. VII., případně metod VI. a VIII.

Sumační metoda I.		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	33,2166	1,0000
II.	18,7301	1,7734
III.	14,1944	2,3401
IV.	24,5397	1,3536
V.	19,4458	1,7082
VI.	12,6476	2,6263
VII.	19,9188	1,6676
VIII.	18,2126	1,8238
IX.	17,0687	1,9461
X.	0,8847	37,5456

Tab. 13. Vyhodnocení Sumační metody I. pro všechny HW konfigurace

HW konfigurace č. X. je založena na výpočtu CAF pomocí grafické karty (GPU), výpočet je založen na využití všech 512 jader GPU, která procesor obsahuje. Pro

využití GPU pro výpočet *CAF* bylo nutné kompletně přeprogramovat implementaci algoritmu dané metody. Největší obtíže byly způsobeny omezenou velikostí paměti na grafické kartě (3 GB vers. 12 GB pro Multi-core CPU). Z tohoto důvodu bylo nutné rozdělit výpočet pomocí GPU na několik bloků (pomocí *for-end* cyklu), jejichž počet byl volen tak, aby vždy došlo k maximálnímu využití paměti na grafické kartě. Dále bylo velice důležité zamezit komunikaci mezi pamětí RAM a pamětí GPU, kde tato komunikace zabírá velkou procentuální část doby výpočtu. Po optimalizaci algoritmu je výsledné zrychlení 38x násobné, což považuji za velice kladný výsledek, a to zejména z důvodů, že urychlení výpočetní doby algoritmu více než desetinásobné má již zcela zásadní dopad na efektivitu implementace a kromě jiného i na finanční náklady na použité finální řešení. Stejného urychlení pomocí CPU by bylo dosaženo až při počtu jader v řádu 48 až 64, což odpovídá nutnosti pořízení několika nových pracovních stanic.

7.3.3 *Sumační metoda II.*

Různorodost optimalizačních technik je dobře viditelná na porovnání metod Sumační metoda I. a Sumační metoda II. Obě metody jsou založené na shodném principu, rozdílnost je dána odlišnou optimalizací. V první metodě je výpočet tvořen dvěma *for-end* cykly, v případě druhé metody, je druhý cyklus nahrazen vektorizovanou verzí výpočtu (maticovými operacemi), se kterou umí systém Matlab pracovat výrazně efektivněji. Jak lze vidět z naměřených hodnot uvedených v Tab. 14, v případě sériového zpracování dat je doba výpočtu téměř trojnásobně nižší, než v případě první metody uvedené v kapitole 7.3.2.

Metoda SMII										
HW konfigurace	Sériové zpracování		Paralelní zpracování							GPU zpracování
	Sériové zpracování	Multi-Core (1 PC stanice)	Cluster (více PC stanic)							
Číslo měření	I.	II.	III.	IV.	V.	VI.	VII.	VIII.	IX.	X.
1	12,4255	12,1871	11,2785	11,2300	13,3277	10,0108	13,3542	11,3291	13,2215	0,8794
2	12,3739	11,9121	11,0994	11,6215	13,4670	10,0994	13,9363	11,4125	11,9374	0,8867
3	16,8506	11,7280	11,2349	14,1422	12,8575	13,0162	17,6458	11,6815	11,9127	0,8849
4	12,3798	11,9216	11,2792	11,7077	13,7089	12,5617	11,6100	11,1544	12,5619	0,8830
5	12,4115	11,5744	11,2873	15,5305	12,7376	10,1059	13,8707	11,3044	11,6092	0,8869
6	12,4322	11,5812	11,3019	14,1475	13,5833	13,2124	13,4017	12,7387	11,9001	0,8852
7	12,3997	11,6078	11,3019	16,5019	13,4333	13,0546	17,4124	11,1593	12,4644	0,8862
8	12,3940	11,6531	11,3862	18,5659	13,3701	10,0463	11,5525	11,3145	13,2676	0,8847
9	12,4133	11,6182	11,3028	13,9850	13,8447	10,1533	13,5200	12,4070	12,7851	0,8864
10	16,8257	11,6400	11,3315	14,2014	10,6633	12,9037	14,0377	16,8945	11,9809	0,8854
11	16,7134	11,6279	11,2827	18,4999	10,5717	10,0588	13,5235	11,3871	12,5730	0,8824
12	12,3546	14,1028	14,4376	18,4991	10,6062	13,0733	11,5726	12,6204	11,6797	0,8864
13	12,3609	11,6164	11,2524	14,3011	10,5257	10,0793	13,5041	12,7450	12,9489	0,8838
14	12,3640	11,6106	11,2044	14,0089	10,6536	10,0668	11,6301	11,6210	11,8913	0,8792
15	12,6264	11,6513	11,2326	11,3951	13,2147	12,3093	11,6392	11,3752	11,8080	0,8834
16	12,2739	11,6225	11,2535	18,6286	13,2706	10,0618	12,3983	11,4289	12,5046	0,8862
17	12,3682	11,6116	13,6998	18,5253	13,8934	13,3213	13,8097	12,0231	11,9762	0,8839
18	12,3812	11,6459	11,2711	14,1942	10,5641	10,1012	11,5095	11,7370	12,6001	0,8885
19	12,3627	14,4617	11,1848	14,5080	10,5242	12,3743	11,5104	11,6511	11,8113	0,8854
20	12,3587	11,6729	11,1978	14,7633	13,2631	10,0204	14,0006	12,0413	11,8667	0,8844
Aritmetický průměr	13,0535	11,9524	11,5410	14,9479	12,4040	11,3315	13,2720	12,0013	12,2650	0,8847
Minimální doba výpočtu	12,2739	11,5744	11,0994	11,2300	10,5242	10,0108	11,5095	11,1544	11,6092	0,8792
Maximální doba výpočtu	16,8506	14,4617	14,4376	18,6286	13,8934	13,3213	17,6458	16,8945	13,2676	0,8885

Tab. 14. Vyhodnocení Sumační metody pro všechny HW konfigurace

Díky paralelizaci jednoho ze dvou cyklů v algoritmu, byla významně urychlena doba výpočtu. Srovnáním dob výpočtu pro jednotlivé HW konfigurace dojdeme ke zjištění, že další urychlení s rostoucím počtem výpočetních jader již nenastává, a to zejména z důvodu velkého množství dat, jejichž zpracování již neumožňuje další urychlení. Obecně lze říci, že s nárůstem velikosti zpracovávaných dat klesá urychlení výpočtu (výpočetní doba na každém jádru je velice krátká k poměru doby potřebné k přesunu dat pro výpočet do jednotlivých jader, tzv. Data transfer CPU-RAM overhead).

Algoritmus pro GPU zpracování je shodný s algoritmem uvedeným v Sumační metodě I., a to z důvodu odlišnosti programování algoritmu pro GPU. V GPU verzi nelze od sebe odlišit algoritmy výpočtu CAF pomocí SMI a SMII. Ze souhrnné Tab. 15, je opět vidět převaha GPU verze oproti ostatním HW konfiguracím.

Sumační metoda II.		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	13,0535	1,0000
II.	11,9524	1,0396
III.	11,5410	1,0766
IV.	14,9479	0,8313
V.	12,4040	1,0017
VI.	11,3315	1,0965
VII.	13,2720	0,9362
VIII.	12,0013	1,0353
IX.	12,2650	1,0131
X.	0,8847	14,0449

Tab. 15. Vyhodnocení Sumační metody II. pro všechny HW konfigurace

7.3.4 Korelační metoda

Korelační metoda výpočtu, jak již z názvu vyplývá, je založena na výpočtu vzájemné korelační funkce, jež ve výpočetním systému MATLAB využívá FFT pro optimalizaci výpočtu. Jedná se o nejpomalejší metodu výpočtu CAF ze všech uvedených v této práci kromě metody SMD. Průměrná doba výpočtu u sériového zpracování dosahuje 99,1 s. S růstem počtu výpočetních jader lze vyvodit závěry uvedené pro sumační metodu I, tzn. zpracování Multi-Core řešení dosahuje odpovídající úrovně očekávaného zrychlení výpočtu (2 až 3,3x). Shodné závěry platí i pro výpočty pomocí clusteru, kde opět díky nárůstu komunikace mezi jednotlivými výpočetními uzly dochází k nárůstu „režijní“ komunikace a zrychlení již nedosahuje očekávaných hodnot. Naměřené výsledky jsou uvedené v Tab. 16. Je nutno poznamenat, že korelační metoda, je extrémně náročná na velikost paměti RAM, kde pro zpracování

signálu o 200.000 vzorcích bylo nutné mít na každém výpočetním uzlu 12 GB operační paměti RAM. Při menší velikosti paměti již dojde ke swapování dat (ukládání mezivýsledků z operační paměti na HDD), a tím k nárůstu doby výpočtu až na úroveň hodnot 5 až 7 minut.

METODA XC												
HW konfigurace ----- Číslo měření	Sériové zpracování		Paralelní zpracování									
	I.	Multi-Core (1 PC stanice)		Cluster (více PC stanic)							GPU zpracování	
		II.	III.	IV.	V.	VI.	VII.	VIII.	IX.	X.		
1	98,9846	48,5179	30,3940	51,3550	26,6130	31,6525	31,6525	24,1809	22,4129	1,4476		
2	99,1278	48,2346	30,4412	50,1027	27,2479	31,9767	31,9767	26,0676	21,6723	1,4502		
3	98,8769	48,4410	29,6875	48,1486	32,9497	35,4207	35,4207	29,9530	20,9899	1,4475		
4	98,8303	48,3562	30,6545	53,4927	29,4682	35,8567	36,8567	26,0000	21,1172	1,4477		
5	99,2308	48,5536	30,6444	50,4159	26,5923	33,3056	33,3056	27,3886	21,5084	1,4471		
6	99,3061	48,0708	31,0599	54,7791	31,0130	32,1031	32,1031	28,7176	21,0593	1,4502		
7	98,9987	48,4846	30,9940	47,4808	32,2488	31,9780	31,9758	29,4250	21,1585	1,4456		
8	98,7861	49,5265	31,7455	50,4144	26,5081	31,1051	31,1051	29,5245	21,4803	1,4488		
9	98,8227	48,5709	30,7857	52,9219	27,2254	32,1592	31,1592	28,0674	21,7827	1,4438		
10	99,1552	48,4528	30,1222	50,1270	27,1463	34,8753	34,8753	23,6576	22,0077	1,4505		
11	99,4038	48,4715	30,5122	53,5794	31,1468	32,1539	32,1539	25,3606	21,2225	1,4434		
12	99,1172	48,7596	30,8072	54,0144	32,1027	32,0586	32,0586	23,7324	21,3388	1,4474		
13	99,1540	48,6483	30,8352	52,8473	32,0112	31,1699	31,1699	24,3532	22,2679	1,4485		
14	99,2305	49,2222	30,2218	55,7444	31,6428	33,6426	33,6426	25,3290	21,9569	1,4480		
15	99,3820	48,5316	30,9399	50,6996	30,5532	31,9958	31,9958	27,5780	21,9395	1,4447		
16	99,1370	48,6278	30,8829	50,5967	28,5266	35,3835	36,3835	24,7645	22,4455	1,4503		
17	99,2879	48,5395	30,2590	52,0415	29,5215	33,2396	33,2396	29,9901	21,7488	1,4475		
18	99,1209	48,4752	31,6520	52,9833	30,5232	35,8814	35,8814	25,2206	21,6616	1,4560		
19	99,2389	48,6197	30,9282	48,4664	29,9967	32,5899	32,5899	25,3178	21,6850	1,4533		
20	99,4546	48,5642	30,7226	49,7859	30,8320	32,1278	32,1278	26,7449	21,8388	1,4497		
Aritmetický průměr	99,1323	48,5834	30,7145	51,4999	29,6935	33,0338	33,0338	26,5687	21,6647	1,4484		
Minimální doba výpočtu	98,7861	48,0708	29,6875	47,4808	26,5081	31,1051	31,1051	23,6576	20,9899	1,4434		
Maximální doba výpočtu	99,4546	49,5265	31,7455	55,7444	32,9497	35,8814	36,8567	29,9901	22,4455	1,4560		

Tab. 16. Vyhodnocení Korelační metody pro všechny HW konfigurace

GPU verze algoritmu pro výpočet *CAF* dosahuje opětovně nejlepších výsledků, kdy zrychlení výpočtu (viz. souhrnná Tab. 17) dosahuje přibližně 68x zrychlení oproti sériovému zpracování. Algoritmus korelační metody, jak již bylo řečeno, klade velké nároky na velikost paměti. Z tohoto důvodu bylo nutné provést výpočet po blocích dat pomocí dvou vnořených cyklů, z nichž vnořený cyklus, obsahující jádro algoritmu, se počítal paralelním způsobem. Struktura výpočtu byla následující:

```

for i=1:ll
    pomocné výpočty
    gfor
        jádro výpočtu
    gend
end

```

Cyklus *gfor-gend* je označení paralelní verze cyklu využívající pro výpočet GPU karty pomocí softwaru Accelereyes Jacket (GPU Toolbox pro prostředí Matlab) použitého pro tuto práci.

Korelační metoda		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	99,1323	1,0000
II.	48,5834	2,0405
III.	30,7145	3,2275
IV.	51,4999	1,9249
V.	29,6935	3,3385
VI.	33,0338	3,0009
VII.	33,0837	2,9964
VIII.	26,5687	3,7312
IX.	21,6647	4,5758
X.	1,4484	68,4426

Tab. 17. Souhrnné vyhodnocení korelační metody pro všechny HW konfigurace

7.3.5 FFT metoda

Algoritmus založený na výpočtu *CAF* pomocí rychlé Fourierovy transformace je z hlediska použití velice oblíbený, a to zejména díky optimalizaci a rychlému výpočtu FFT metody v mnoha programovacích jazycích. Porovnáním výsledků sériového a Multic-core zpracování, uvedených v Tab. 18, dojdeme k závěru, že jisté zrychlení výpočtu v průběhu zvyšování počtu jader nastává, ale není nikterak významné.

Metoda FFT										
HW konfigurace ----- Číslo měření	Sériové zpracování		Paralelní zpracování							
	I.	Multi-Core (1 PC stanice)		Cluster (více PC stanic)						GPU zpracování
		II.	III.	IV.	V.	VI.	VII.	VIII.	IX.	
1	2,0637	1,3715	1,1672	1,2029	0,8649	0,7461	0,7573	0,6379	0,9078	0,1773
2	2,0630	1,3642	1,2100	1,1608	0,7928	0,7383	0,7374	0,6430	0,8617	0,1768
3	2,0565	1,3577	1,1707	1,1571	0,8053	0,7282	0,7327	0,6553	0,8784	0,1756
4	2,0600	1,3661	1,1733	1,1538	0,8118	0,7331	0,7344	0,6229	0,8698	0,1754
5	2,0653	1,3652	1,1678	1,1533	0,8169	0,7357	0,7565	0,6531	0,8877	0,1759
6	2,0612	1,3695	1,1687	1,1569	0,8136	0,7412	0,7657	0,6446	0,9273	0,1758
7	2,0599	1,3667	1,1648	1,1530	0,8157	0,7420	0,7363	0,6485	0,8574	0,1758
8	2,0628	1,3743	1,1693	1,1607	0,7901	0,7414	0,745	0,6726	0,8910	0,1760
9	2,0609	1,3651	1,1766	1,1653	0,8033	0,7490	0,7334	0,6297	0,8487	0,1756
10	2,0600	1,3708	1,1628	1,1556	0,8155	0,7397	0,7308	0,6380	0,8804	0,1753
11	2,0580	1,6640	1,1821	1,1691	0,7997	0,7206	0,7533	0,6564	0,8604	0,1744
12	2,0551	1,3659	1,1748	1,1627	0,8177	0,7590	0,7668	0,6432	0,8877	0,1750
13	2,0628	1,3713	1,1645	1,1538	0,8038	0,7448	0,7442	0,6241	0,8739	0,1762
14	2,0592	1,3639	1,1468	1,1638	0,8232	0,7481	0,7342	0,6446	1,0130	0,1762
15	2,0603	1,3326	1,1656	1,1550	0,7944	0,7239	0,7269	0,6701	0,9071	0,1751
16	2,0594	1,3699	1,1618	1,1571	0,8216	0,7686	0,7525	0,6361	1,0386	0,1758
17	2,0594	1,3688	1,1599	1,1689	0,7970	0,7273	0,7466	0,6220	0,8642	0,1755
18	2,0612	1,3700	1,1702	1,1603	0,8101	0,7444	0,7625	0,6451	0,8827	0,1755
19	2,0611	1,3746	1,1689	1,1541	0,8168	0,7310	0,7337	0,6567	0,8718	0,1756
20	2,0586	1,3302	1,1568	1,1566	0,8072	0,7840	0,752	0,6259	0,8846	0,1750
Aritmetický průměr	2,0604	1,3791	1,1691	1,1610	0,8111	0,7423	0,7451	0,6435	0,8947	0,1757
Minimální doba výpočtu	2,0551	1,3302	1,1468	1,1530	0,7901	0,7206	0,7269	0,6220	0,8487	0,1744
Maximální doba výpočtu	2,0653	1,6640	1,2100	1,2029	0,8649	0,7840	0,7668	0,6726	1,0386	0,1773

Tab. 18. Vyhodnocení FFT metody pro všechny HW konfigurace

Důvodem je příliš krátká doba vlastního výpočtu, velkou část doby zabere komunikace způsobená paralelním zpracováním dat [67], [68] a také samotná filozofie produktu MATLAB, která je postavena na premise, že paralelní zpracování dat je vhodné a žádoucí zejména v případech, kde výpočet trvá od desítek sekund až po stovky hodin. Z tohoto důvodu režijní doba komunikace v řádu jednotek sekund, ve které je zahrnuto spuštění nadstavby pro paralelní zpracování, komunikace mezi jádry, atd., je v poměru k době zpracování vlastního výpočtu velice malá. K významné úspoře času dochází až při splnění podmínky, že doba výpočtu je v řádu desítek sekund. Tento předpoklad potvrzují i naměřené údaje uvedené v souhrnné Tab. 19, kde pro HW konfiguraci IX, tj. s největším počtem výpočetních jader, dochází dokonce k nárůstu doby výpočtu oproti metodám obsahující nižší počet výpočetních jader.

FFT metoda		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	2,0604	1,0000
II.	1,3791	1,4940
III.	1,1691	1,7624
IV.	1,1610	1,7747
V.	0,8111	2,5403
VI.	0,7423	2,7757
VII.	0,7451	2,7653
VIII.	0,6435	3,2019
IX.	0,8947	2,3029
X.	0,1757	11,7268

Tab. 19. Souhrnné vyhodnocení FFT metody pro všechny HW konfigurace

K významnému zrychlení výpočtu došlo opět v algoritmu pro GPU verzi, kde i přes relativně nízkou hodnotu sériového zpracování (cca 2s) došlo téměř k 12-ti násobnému urychlení doby výpočtu.

7.3.6 Maticová metoda výpočtu CAF

Maticová metoda výpočtu CAF stejně s metodou výpočtu ve frekvenční oblasti je založena na plně maticové formulaci výpočtu. Rozdíl obou metod je především v odlišnosti v práci se vstupními signály, kde maticová metoda počítá s celou délkou přímého a odraženého signálu. Při hodnotách uvedených v Tab. 11 se jedná o práci s maticemi obsahujícími cca $40 \cdot 10^6$ prvků komplexních čísel, což klade extrémní nároky na velikost operační paměti RAM. Naměřené hodnoty uvedené v Tab. 20 nevykazují rozdíly v sériovém, Multi-core ani clusterovém zpracování, což je

způsobeno právě extrémními nároky na velikost paměti, a to díky výše uvedené velikosti matic pro výpočet.

Metoda MM														
HW konfigurace ----- Číslo měření	Sériové zpracování		Paralelní zpracování									GPU zpracování		
	I.	Multi-Core (1 PC stanice)	Cluster (více PC stanic)									X. P=20	XI. P=10	XII. P=5
			II.	III.	IV.	V.	VI.	VII.	VIII.	IX.				
1	9,6116	9,6046	9,7330	9,2215	9,7651	9,6621	9,6702	9,4671	9,5332	1,9455	1,3284	0,9514		
2	9,3256	9,5677	9,5213	9,7848	9,4527	9,4702	9,5408	9,4599	9,6064	1,9438	1,3245	0,9527		
3	9,4906	9,5740	9,5498	9,5513	9,5062	9,5779	9,4091	9,4022	9,5224	1,9409	1,3263	0,9530		
4	9,4097	9,5130	9,5596	9,5168	9,6100	9,5138	9,3984	9,5472	9,4869	1,9430	1,3272	0,9795		
5	9,4963	9,5657	9,6639	9,6204	9,3939	9,5314	9,4734	9,5022	9,5059	1,9425	1,3311	0,9522		
6	9,5828	9,4072	9,5842	9,5243	9,4542	9,5439	9,5530	9,4318	9,6210	1,9416	1,3295	0,9528		
7	9,5833	9,5411	9,4423	9,3982	9,5464	9,5555	9,5763	9,6269	9,5287	1,9429	1,3251	0,9993		
8	9,5696	9,5225	9,5478	9,4946	9,5325	9,5261	9,3933	9,0805	9,4817	1,9405	1,3277	0,9779		
9	9,6061	9,5505	9,6209	9,6322	9,6053	9,5394	9,4207	9,6176	9,5243	1,9416	1,3295	0,9535		
10	9,6046	9,5626	9,5871	9,4935	9,4294	9,5200	9,4487	9,5503	9,5226	1,9422	1,3285	0,9925		
11	9,6514	9,5556	9,5646	9,5976	9,5127	9,5361	9,5772	9,1869	9,5314	1,9418	1,3274	0,9537		
12	9,5542	9,6017	9,6004	9,5996	9,3878	9,5485	9,5729	9,4893	9,4685	1,9449	1,3276	0,9520		
13	9,5326	9,5922	9,5085	9,5930	9,4889	9,5918	9,4212	9,1515	9,6068	1,9423	1,3266	0,9521		
14	9,5752	9,4509	9,6360	9,5629	9,5224	9,5580	9,5069	9,8002	9,5341	1,9419	1,3291	0,9924		
15	9,5060	9,6217	9,6110	9,5528	9,5313	9,5520	9,4225	9,4395	9,5229	1,9413	1,3281	0,9807		
16	9,4964	9,5265	9,5526	9,5623	9,5422	9,4987	9,4377	9,4314	9,5510	1,9423	1,3290	0,9528		
17	9,4434	9,5627	9,6856	9,4711	9,4043	9,5025	9,5839	9,5099	9,6731	1,9440	1,3286	0,9887		
18	9,6130	9,5612	9,5883	9,5716	9,7059	9,5251	9,4001	9,3949	9,4965	1,9411	1,3263	0,9528		
19	9,4880	9,6377	9,5040	9,6204	9,4955	9,4516	9,5057	9,4580	9,4933	1,9421	1,3273	1,0065		
20	9,5534	9,5063	9,5971	9,5311	9,4846	9,4324	9,5332	9,4829	9,4232	1,9417	1,3280	0,9878		
Aritmetický průměr	9,5347	9,5513	9,5829	9,5450	9,5186	9,5319	9,4923	9,4515	9,5317	1,9424	1,3278	0,9515		
Minimální doba výpočtu	9,3256	9,4072	9,4423	9,2215	9,3878	9,4324	9,3933	9,0805	9,4232	1,9405	1,3245	0,9514		
Maximální doba výpočtu	9,6514	9,6377	9,7330	9,7848	9,7651	9,6621	9,6702	9,8002	9,6731	1,9455	1,3311	1,0065		

Tab. 20. Vyhodnocení maticové metody pro všechny HW konfigurace

Algoritmus pro GPU verzi výpočtu bylo nutné rozdělit pomocí cyklu do několika částečných výpočtů (bloků), a to z důvodu omezení velikosti paměti na GPU (3 GB). Vliv počtu bloků, do kterých je výpočet rozdělen, významně ovlivní dobu výpočtu, jak je patrné ze souhrné Tab. 21 řádky 10-12. Při rozdělení do dvaceti bloků dat nedochází k úplnému využití volné paměti grafické karty a k nárůstu počtu iterací cyklu (počet bloků, do kterých je rozdělen výpočet, je proveden pomocí *for-end* cyklu). Se snižujícím se počtem bloků dojde k významnému urychlení výpočtu až na úroveň 10-ti násobného zrychlení pro 5 bloků dat. Nižší počet bloků již není možné stanovit z důvodu nedostatku paměti (čím nižší počet bloků, tím je větší velikost každého bloku).

Metoda výpočtu pomocí maticové metody		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	9,5347	1,0000
II.	9,5513	0,9983
III.	9,5829	0,9950
IV.	9,5450	0,9989
V.	9,5186	1,0017
VI.	9,5319	1,0003
VII.	9,4923	1,0045
VIII.	9,4515	1,0088
IX.	9,5317	1,0003
X. (P=20)	1,9424	4,9087
XI. (P=10)	1,3278	7,1808
XII. (P=5)	0,9515	10,0207

Tab. 21. Souhrnné vyhodnocení FMCW metody pro všechny HW konfigurace

7.3.7 FFT metoda s decimací signálového součinu

Tato metoda je založená na velice podobném principu jako FFT metoda s rozdílem, že před vlastním výpočtem *CAF* dojde k decimaci vstupních dat, a tím k redukci náročnosti výpočtu (výsledky v Tab. 22). Se vzrůstajícím faktorem decimace dochází k vyšší redukci vstupních dat, a tím k nižší výpočetní náročnosti. Faktor decimace byl zvolen $L=4$. Z důvodu koherentního zpracování dat lze uvažovat maximálně faktor decimace $L=8$. Vyšší decimační faktory nelze doporučit z důvodu významného poškození fázové informace.

Metoda FFTD												
HW konfigurace ----- Číslo měření	Sériové zpracování		Paralelní zpracování									
	I.	Multi-Core (1 PC stanice)		Cluster (více PC stanic)							GPU zpracování	
		II.	III.	IV.	V.	VI.	VII.	VIII.	IX.	X.	XI.	
1	1,7481	1,4058	1,3175	1,3406	1,1233	1,1104	1,3218	1,0812	1,4800	0,1782	0,1366	
2	1,7347	1,3921	1,3113	1,3414	1,0948	1,0257	1,2011	1,0912	1,7302	0,1778	0,1378	
3	1,7306	1,3489	1,3101	1,3435	1,1019	1,0134	1,1574	1,0891	1,5133	0,1777	0,1370	
4	1,7403	1,3768	1,3218	1,3431	1,0977	1,0476	1,1579	1,1406	1,5048	0,1777	0,1371	
5	1,7252	1,3385	1,2989	1,3283	1,0843	1,1081	1,5503	1,1839	1,4935	0,1771	0,1370	
6	1,7229	1,3939	1,2699	1,3369	1,1175	1,0658	1,6840	1,0822	1,4709	0,1776	0,1362	
7	1,7253	1,3725	1,2944	1,3451	1,0667	1,0532	1,1899	1,1809	1,4841	0,1781	0,1373	
8	1,7290	1,3888	1,2834	1,3420	1,0893	1,0864	1,2054	1,5107	1,7084	0,1774	0,1371	
9	1,7361	1,3897	1,3141	1,3228	1,0882	1,0749	1,3148	1,2356	1,4976	0,1776	0,1374	
10	1,7758	1,3866	1,2608	1,3243	1,0958	1,0982	1,3918	1,1215	1,4800	0,1778	0,1371	
11	1,7343	1,3871	1,2706	1,3423	1,0737	1,0769	1,3260	1,1215	1,7570	0,1778	0,1375	
12	1,7326	1,3861	1,2699	1,3223	1,0841	1,0685	1,2890	1,5201	1,5052	0,1783	0,1369	
13	1,7349	1,3810	1,2644	1,3415	1,1008	1,0992	1,2145	1,1836	1,5100	0,1774	0,1378	
14	1,7608	1,3871	1,3419	1,7197	1,0854	1,0775	1,1574	1,2640	1,4724	0,1780	0,1373	
15	1,7489	1,3711	1,3408	1,3206	1,0901	1,1017	1,2190	1,1887	1,4797	0,1775	0,1368	
16	1,7446	1,3794	1,2846	1,3320	1,1088	0,9789	1,2032	1,3643	1,5509	0,1782	0,1370	
17	1,7355	1,3813	1,2848	1,3475	1,0736	1,0022	1,2195	1,3851	1,5827	0,1780	0,1371	
18	1,7335	1,3448	1,2898	1,3489	1,0899	1,0486	1,2118	1,2258	1,4908	0,1777	0,1360	
19	1,7392	1,3795	1,2636	1,3273	1,0981	1,0727	1,1737	1,2092	1,5544	0,1780	0,1381	
20	1,7470	1,3975	1,3139	1,3212	1,0763	1,0536	1,8325	1,0670	1,4836	0,1777	0,1375	
Aritmetický průměr	1,7390	1,3794	1,2953	1,3546	1,0920	1,0632	1,3011	1,2123	1,5375	0,1778	0,1371	
Minimální doba výpočtu	1,7229	1,3385	1,2608	1,3206	1,0667	0,9789	1,1574	1,0670	1,4709	0,1771	0,1360	
Maximální doba výpočtu	1,7758	1,4058	1,3419	1,7197	1,1233	1,1104	1,8325	1,5201	1,7570	0,1783	0,1381	

Tab. 22. Vyhodnocení FFTD metody pro všechny HW konfigurace

FFT metoda s decimací signálového součinu		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	1,7390	1,0000
II.	1,3794	1,2607
III.	1,2953	1,3425
IV.	1,3546	1,2838
V.	1,0920	1,5925
VI.	1,0663	1,6308
VII.	1,3011	1,3366
VIII.	1,2123	1,4345
IX.	1,5375	1,1311
X.	0,1371	12,6842

Tab. 23. Souhrnné vyhodnocení FFTD metody pro všechny HW konfigurace

Algoritmus výpočtu této metody se od předchozí metody FFT odlišuje pouze vlastní decimací signálu. Z důvodu přidání příkazů pro decimaci (příkazy reshape a následovné násobení matic) a příliš nízké doby vlastního výpočtu (v řádu jednotek sekund) nedojde k urychlení doby výpočtu, ale dokonce k jistému prodloužení této doby. Pro vyšší faktory decimace ($L=16, 32, 64, 128$) již dochází k urychlení výpočtu, ale z výše uvedeného důvodu poškození fázové informace nebyly tyto výpočty zahrnuty do výsledných tabulek.

Nejlepších výsledků opět dosáhl algoritmus výpočtu pomocí GPU, kde došlo téměř k 13-ti násobnému urychlení doby výpočtu (viz Tab. 23).

7.3.8 Metoda výpočtu CAF ve frekvenční oblasti

Princip algoritmu této metody je podrobně popsán v Tab. 7, kde je výpočet založen na dvou vnořených cyklech. Z důvodu paralelizace výpočtu bylo nutné tyto cykly od sebe oddělit, či případně úplně odstranit (kompletní vektorizace kódu do maticového zpracování). Tato kompletní vektorizace kódu byla úspěšně provedena, čímž došlo k významnému urychlení vlastního výpočtu CAF, kde i pro sériové zpracování byl využit algoritmus pouze pomocí maticového počtu. Průměrná doba výpočtu v sériovém zpracování je dána hodnotou 0,05 s. Díky krátké době výpočtu již v sériové verzi nebylo urychlení doby výpočtu pomocí Multi-core a clusteru již téměř rozpoznatelné, což je způsobeno shodnými důvody, které jsou uvedené v kapitole Tab. 21. Naměřené hodnoty jsou zobrazeny v Tab. 24.

Metoda FMGFD															
		Sériové zpracování		Paralelní zpracování											
HW konfigurace		Sériové zpracování		Multi-Core (1 PC stanice)		Cluster (více PC stanice)								GPU zpracování	
----- Číslo měření		I.	II.	III.	IV.	V.	VI.	VII.	VIII.	IX.	X. L=200				
1		0,0715	0,0568	0,0507	0,0497	0,0572	0,0503	0,0505	0,0509	0,0599	0,0506				
2		0,0493	0,0498	0,0530	0,0501	0,0508	0,0507	0,0507	0,0508	0,0602	0,0478				
3		0,0495	0,0506	0,0499	0,0503	0,0505	0,0512	0,0509	0,0509	0,0602	0,0487				
4		0,0496	0,0502	0,0498	0,0502	0,0508	0,0509	0,0508	0,0503	0,0603	0,0492				
5		0,0494	0,0506	0,0501	0,0503	0,0502	0,0525	0,0509	0,0507	0,0600	0,0489				
6		0,0494	0,0500	0,0604	0,0501	0,0507	0,0507	0,0549	0,0518	0,0599	0,0505				
7		0,0493	0,0504	0,0500	0,0502	0,0502	0,0501	0,0510	0,0512	0,0597	0,0503				
8		0,0496	0,0505	0,0503	0,0503	0,0507	0,0505	0,0517	0,0507	0,0592	0,0524				
9		0,0488	0,0504	0,0502	0,0506	0,0506	0,0504	0,0508	0,0505	0,0596	0,0487				
10		0,0491	0,0498	0,0504	0,0504	0,0505	0,0501	0,0509	0,0503	0,0597	0,0527				
11		0,0498	0,0500	0,0507	0,0501	0,0510	0,0508	0,0506	0,0505	0,0599	0,0512				
12		0,0496	0,0506	0,0505	0,0501	0,0531	0,0509	0,0509	0,0503	0,0600	0,0507				
13		0,0498	0,0505	0,0500	0,0499	0,0504	0,0509	0,0509	0,0513	0,0595	0,0515				
14		0,0493	0,0500	0,0506	0,0499	0,0504	0,0506	0,0511	0,0511	0,0604	0,0476				
15		0,0515	0,0509	0,0507	0,0507	0,0510	0,0506	0,0508	0,0509	0,0595	0,0501				
16		0,0497	0,0497	0,0501	0,0541	0,0504	0,0506	0,0515	0,0506	0,0604	0,0477				
17		0,0499	0,0499	0,0505	0,0502	0,0510	0,0522	0,0508	0,0504	0,0598	0,0522				
18		0,0496	0,0502	0,0505	0,0500	0,0509	0,0505	0,0508	0,0518	0,0595	0,0425				
19		0,0501	0,0502	0,0506	0,0503	0,0505	0,0506	0,0510	0,0502	0,0598	0,0508				
20		0,0495	0,0504	0,0504	0,0506	0,0507	0,0509	0,0523	0,0501	0,0605	0,0516				
Aritmetický průměr		0,0507	0,0506	0,0510	0,0504	0,0511	0,0508	0,0512	0,0508	0,0599	0,0498				
Minimální doba výpočtu		0,0488	0,0497	0,0498	0,0497	0,0502	0,0501	0,0505	0,0501	0,0592	0,0425				
Maximální doba výpočtu		0,0715	0,0568	0,0604	0,0541	0,0572	0,0525	0,0549	0,0518	0,0605	0,0527				

Tab. 24. Vyhodnocení FMGFD metody pro všechny HW konfigurace

GPU zpracování vykazuje jisté zrychlení, ale z důvodu již, tak rychlých časů v sériovém zpracování, není tento zisk nijak významný. Za úvahu stojí možnost zvýšení celkové délky signálu a vzorkovací frekvence, což by vedlo k zvýšení rozlišení jak v bistatické vzdálenosti, tak i v Dopplerově posunu. Souhrnná tabulka této metody je uvedena v Tab. 25.

Metoda výpočtu ve frekvenční oblasti		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	0,0507	1,0000
II.	0,0506	1,0020
III.	0,0510	0,9941
IV.	0,0504	1,0060
V.	0,0511	0,9922
VI.	0,0508	0,9980
VII.	0,0512	0,9902
VIII.	0,0508	0,9980
IX.	0,0599	0,8464
X.	0,0498	1,0181

Tab. 25. Souhrnné vyhodnocení FMGFD metody pro všechny HW konfigurace

7.3.9 Metoda výpočtu CAF pomocí částečných korelací

Jedná se o metodu výpočtu CAF pomocí rozdělení vstupního a odraženého signálu do bloků různých délek, ze kterých je pak vypočítána vzájemná korelace. Díky tomuto rozdělení je výpočet vzájemné korelace podstatně méně náročný než v případě výpočtu na celé délce signálů. Díky příliš krátké době výpočtu se projeví urychlení doby výpočtu pouze na Multi-core verzi (zejména při plném vytížení 4 jader – HW konfigurace III.), což je způsobeno nárůstem „režie“ potřebné pro paralelní výpočet. Zrychlení GPU verze algoritmu je téměř shodné s HW konfigurací III., což je způsobeno použitým *for-end* cyklem, jenž vytváří jednotlivé bloky signálů, ale s přesahem hodnot jednotlivých bloků. Toto vytváření bloků se bohužel nepodařilo kompletně vektorizovat či převést na paralelní verzi cyklu. Doba výpočtu těchto bloků tvoří přitom cca 70% času výpočetní doby celé metody. V případě paralelizace by metoda dosahovala teoreticky času na úrovni 0,15 s. Níže jsou v Tab. 26 a Tab. 27 uvedeny všechny naměřené a souhrnné hodnoty pro tuto metodu.

Metoda FMCW										
HW konfigurace ----- Číslo měření	Sériové zpracování		Paralelní zpracování							
	I.	Multi-Core (1 PC stanice)		Cluster (více PC stanic)						GPU zpracování
		II.	III.	IV.	V.	VI.	VII.	VIII.	IX.	
1	0,7486	0,4806	0,4082	0,5515	0,5030	0,5659	0,8589	1,2184	1,7730	0,4565
2	0,7436	0,4666	0,4001	0,5651	0,4952	0,5579	1,1500	1,2483	1,8067	0,4717
3	0,7390	0,4689	0,3955	0,5783	0,4803	0,5559	0,8734	0,9670	1,5728	0,4852
4	0,7479	0,4924	0,3926	0,5848	0,4641	0,5610	0,8641	1,2405	1,7739	0,4502
5	0,7409	0,4647	0,4008	0,5665	0,4830	0,5453	0,8813	1,2294	1,7965	0,4875
6	0,7411	0,4701	0,3927	0,5517	0,5047	0,5864	1,1391	1,2499	1,5813	0,4965
7	0,7372	0,4695	0,4036	0,5804	0,4760	0,5492	1,1277	1,2591	1,7533	0,5003
8	0,7423	0,4655	0,3970	0,5558	0,5108	0,5499	1,1315	1,2304	1,7865	0,4623
9	0,7456	0,4684	0,3985	0,5730	0,4962	0,5424	1,1423	0,9675	1,5909	0,4782
10	0,7379	0,4685	0,3954	0,5690	0,4992	0,5450	0,8508	1,2626	1,5902	0,4723
11	0,7391	0,4653	0,3922	0,5727	0,4693	0,5549	1,1095	1,2545	1,7623	0,4632
12	0,7383	0,4677	0,3968	0,5602	0,4876	0,5571	0,8639	1,2368	1,5887	0,4639
13	0,7412	0,4663	0,4034	0,5666	0,4737	0,5740	0,8598	0,9684	1,7837	0,4798
14	0,7377	0,4685	0,4105	0,5706	0,4947	0,5566	0,8657	1,2164	1,5498	0,4909
15	0,7388	0,4654	0,4498	0,5755	0,4739	0,5503	1,1629	1,2316	1,8010	0,4785
16	0,7362	0,4660	0,3904	0,5693	0,4767	0,5450	0,8773	1,2107	2,0032	0,4865
17	0,7410	0,4695	0,3990	0,5648	0,5047	0,5539	0,8799	1,2938	1,6033	0,4839
18	0,7398	0,4643	0,3904	0,5598	0,4904	0,5586	1,0102	1,2217	1,8008	0,4982
19	0,7443	0,4657	0,3909	0,5529	0,4802	0,5472	0,8632	1,2350	1,6006	0,4652
20	0,7429	0,4610	0,3913	0,5662	0,4705	0,5463	0,9856	0,9856	1,7721	0,4500
Aritmetický průměr	0,7412	0,4687	0,4000	0,5667	0,4867	0,5551	0,9749	1,1864	1,7145	0,4760
Minimální doba výpočtu	0,7362	0,4610	0,3904	0,5515	0,4641	0,5424	0,8508	0,9670	1,5498	0,4500
Maximální doba výpočtu	0,7486	0,4924	0,4498	0,5848	0,5108	0,5864	1,1629	1,2938	2,0032	0,5003

Tab. 26. Vyhodnocení FMCW metody pro všechny HW konfigurace

Metoda výpočtu pomocí částečných korelací		
HW konfigurace	Doba výpočtu [s]	Zrychlení [-]
I.	0,7412	1,0000
II.	0,4687	1,5814
III.	0,4000	1,8530
IV.	0,5667	1,3079
V.	0,4867	1,5229
VI.	0,5551	1,3353
VII.	0,9749	0,7603
VIII.	1,1864	0,6247
IX.	1,7145	0,4323
X.	0,4760	1,5703

Tab. 27. Souhrnné vyhodnocení FMCW metody pro všechny HW konfigurace

7.4 Vizuální porovnání vypočtených povrchů CAF pro jednotlivé metody výpočtu

V této kapitole dojde k vyhodnocení a porovnání vypočtených povrchů CAF u všech uvedených implementací, a to z důvodu ověření, že všechny uvedené metody poskytují adekvátní výsledky.

Jako referenční metoda vypočteného povrchu byla opět zvolena sumační metoda výpočtu SMI vypočtená pomocí sériového algoritmu. Princip porovnání jednotlivých metod je založen na výpočtu rozdílu vypočtených povrchů CAF, podle vztahu

$$|CAF_{ROZ}(m, k)| = |CAF_x(m, k)| - |CAF_{REF}(m, k)|, \quad (7.2)$$

kde $CAF_x(m, k)$ je vypočtený povrch CAF pro x-tou metodu,

$CAF_{REF}(m, k)$ je referenční hodnota povrchu.

Seznam rozdílových povrchů s odpovídajícím názvem souboru umístěných na přiloženém DVD je uveden v Tab. 28. Součástí tohoto srovnání, z důvodu odlišnosti výpočetního algoritmu u všech GPU metod, došlo také k jejich porovnání s referenčním povrchem.

Z rozdílových povrchů je zřejmá vysoká shoda u všech testovaných metod, kde průměrná rozdílový povrch dosahuje -50 dB, což považuji za dostatečné ověření přesnosti u všech testovaných metod a potvrzení shody výpočtu.

Název metody	Jméno souboru
SMII	1_SUMII.png
XC	2_XC.png
FFT	3_FFT.png
MM	4_MM.png
FFTD	5_FFTD.png
FMGFD	6_FMGFD.png
FMCW	7_FMCW.png
SM na GPU	8_SUM_GPU.png
XC na GPU	9_XC_GPU.png
FFT na GPU	A_FFT_GPU.png
MM na GPU	B_MM_GPU.png
FFTD na GPU	C_FFTD_GPU.png
FMGFD na GPU	D_FMGFD_GPU.png
FMCW na GPU	E_FMCW_GPU.png

Tab. 28. Seznam rozdílových povrchů

7.5 Celkové vyhodnocení výpočtu CAF napříč všemi metodami

Tato podkapitola je věnována závěrečnému vyhodnocení metod výpočtu *CAF* z hlediska:

- (1) Vlastnosti jednotlivých algoritmů CAF;
- (2) Porovnání z hlediska rychlosti výpočtu pro sériové zpracování;
- (3) Porovnání z hlediska rychlosti výpočtu pro jednotlivé HW konfigurace (zahrnující náročnost na technické vybavení a zprávu zvolené HW konfigurace).

7.5.1 Vlastnosti jednotlivých algoritmů CAF

Každá z uvedených metod výpočtu CAF v této práci má své výhody a nevýhody, které jsou shrnuty v této kapitole. Vlastnosti uvedených metod se netýkají rychlosti vlastního výpočtu CAF, ale spíše možností volby vstupních dat, strukturou algoritmu, náročností na HW vybavení atd. Seznam výhod a nevýhod je sumarizován v Tab. 29.

Metoda	Výhody	Nevýhody
SMI (platí i pro GPU)	<ul style="list-style-type: none"> - Lze volit libovolný počet bistatických vzdáleností i Dopplerových posunů 	<ul style="list-style-type: none"> - Výpočet dle definice obsahuje dva vnořené cykly - Velká výpočetní náročnost
SMII	<ul style="list-style-type: none"> - Lze volit libovolný počet bistatických vzdáleností i Dopplerových posunů - Částečná vektorizace kódu 	<ul style="list-style-type: none"> - K výpočtu použit pouze jeden for-end cyklus
XC	<ul style="list-style-type: none"> - Možnost volby počtu Dopplerových posunů - Dopplerovské posuny nemusejí tvořit celá čísla (libovolné rozlišení v Doppleru) 	<ul style="list-style-type: none"> - Extrémní náročnost na velikost operační paměti RAM - CAF je vypočtena pro všechny možné hodnoty bistatických vzdáleností (interval $\langle -N; N \rangle$)
FFT	<ul style="list-style-type: none"> - Využití FFT pro výpočet (velice rychlé díky vysoké úrovni optimalizace) - Možnost volby počtu bistatických vzdáleností 	<ul style="list-style-type: none"> - CAF je vypočtena pro všechny možné hodnoty Dopplerových posunů (interval $\langle -N/2; N/2 \rangle$) - Dopplerovské posuny musí být celá čísla
FFTD	<ul style="list-style-type: none"> - Možnost volby počtu bistatických vzdáleností - Snížení numerické náročnosti vlivem decimace (má smysl, jen když funguje paralelizace) 	<ul style="list-style-type: none"> - Nemožnost volby počtu Dopplerových posunů - Použití „jedničkového“ filtru
FMGFD	<ul style="list-style-type: none"> - Výpočet bez cyklů (pouze pomocí maticových operací) - Možnost využití libovolného filtru 	<ul style="list-style-type: none"> - Délka a druh filtru ovlivňují rozlišení výsledné CAF

Metoda	Výhody	Nevýhody
FMCW	<ul style="list-style-type: none"> - Velké snížení výpočetní náročnosti - Využití zpracování známého z FMCW zpracování 	<ul style="list-style-type: none"> - Velký vliv volby délky jednotlivých bloků <ul style="list-style-type: none"> o Na úroveň šumu o Rozlišení v bistatické vzdálenosti a Dopplerově posunu
MM	<ul style="list-style-type: none"> - Bez použití cyklů - Možnost volby počtu bistatických vzdáleností a Dopplerových posunů 	<ul style="list-style-type: none"> - Nutnost zpracovávat „velké“ matice - Nevhodné pro cluster díky nárokům na přenos dat po síti

Tab. 29. Shrnutí vybraných parametrů u jednotlivých algoritmů CAF

7.5.2 Porovnání z hlediska rychlosti výpočtu pro sériové zpracování

Z důvodu jednoznačného porovnání metod výpočtu CAF jsou všechny metody vztahované k sumační metodě I (SMI - kapitola 7.3.2), kde celkové naměřené hodnoty s uvedením odpovídajícího zrychlení jsou zobrazeny v Tab. 30 (HW konfigurace I.). Toto srovnání pro sériové zpracování zde bylo uvedeno zejména z důvodu obecného srovnání metod z hlediska rychlosti výpočtu. V kapitole 7.5.3 již dojde pouze k popisu rozdílů jednotlivých HW konfigurací z hlediska rychlosti výpočtu.

Z hlediska porovnání doby výpočtu u *seriového zpracování* je zřejmá kratší doba výpočtu (do cca 2,1 s) u metod pracujících na principu dělení vstupních signálů do bloků dat (FFTD, FMGFD, FMCW). Toto zrychlení je dáno nižší výpočetní náročností jednotlivých matematických operací. Zpravidla (nemusí vždy platit, záleží na konkrétním výpočtu) je výhodnější u velmi dlouhých signálů (v řádu počet vzorků $>10^5$) rozdělit celkový výpočet do opakujících se dílčích výpočtů, které jsou postupně ukládány do celkového výstupu. Významnou výhodou u tohoto postupného výpočtu je daleko nižší náročnost na potřebnou paměť pro výpočet. Například u korelační metody (pracující s celým rozsahem dat) je náročnost na velikost operační paměti RAM v řádu desítek GB oproti metodám založeným na dělení, kde je postačující velikost paměti RAM do 8 GB.

Celkové vyhodnocení metod																				
HW konfigurace	Sériové zpracování		Multi-Core (1 PC stanice)				Cluster (více PC stanic)										GPU zpracování			
	I.		II.		III.		IV.		V.		VI.		VII.		VIII.		IX.		X.	
Název metody	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup	Doba výpočtu	Speedup
Sumační metoda I. (SMI)	33,2166	1,0000	18,7301	1,7734	14,1944	2,3401	24,5397	1,3536	19,4458	1,7082	12,6476	2,6263	19,9188	1,6676	18,2126	1,8238	17,0687	1,9461	0,8847	37,5456
Sumační metoda II. (SMII)	13,0535	2,5447	11,9524	2,7791	11,5410	2,8781	14,9479	2,2222	12,4040	2,6779	11,3315	2,9314	13,2720	2,5028	12,0013	2,7678	12,2650	2,7082	0,8847	37,5456
Korelační metoda (XC)	99,1323	0,3351	48,5834	0,6837	30,7145	1,0815	51,4999	0,6450	29,6935	1,1186	33,0837	1,0040	33,0837	1,0040	26,5687	1,2502	21,6647	1,5332	1,4484	22,9333
FFT metoda (FFT)	2,0604	16,1214	1,3791	24,0857	1,1691	28,4121	1,1610	28,6103	0,8111	40,9525	0,7423	44,7482	0,7451	44,5801	0,6435	51,6186	0,8947	37,1260	0,1757	189,0529
FFT metoda s decimací signálového součinu (FFTD)	1,7390	19,1010	1,3794	24,0805	1,2953	25,6439	1,3546	24,5213	1,0920	30,4181	1,0663	31,1513	1,3011	25,5296	1,2123	27,3997	1,5375	21,6043	0,1371	242,2801
Metoda výpočtu CAF ve frekvenční oblasti (FMGFD)	0,0507	655,1598	0,0506	656,4545	0,0510	651,3059	0,0504	659,0595	0,0511	650,0313	0,0508	653,8701	0,0512	648,7617	0,0508	653,8701	0,0599	554,5342	0,0498	667,0000
Metoda výpočtu CAF pomocí částečných korelací (FMCW)	0,7412	44,8146	0,4687	70,8696	0,4000	83,0415	0,5667	58,6141	0,4867	68,2486	0,5551	59,8389	0,9749	34,0718	1,1864	27,9978	1,7145	19,3739	0,4760	69,7828
Maticová metoda výpočtu CAF (MM)	9,5347	3,4838	9,5513	3,4777	9,5829	3,4662	9,5450	3,4800	9,5186	3,4897	9,5319	3,4848	9,4923	3,4993	9,4515	3,5144	9,5317	3,4849	0,9515 (P=5)	34,9097

Tab. 30. Celkové vyhodnocení metod výpočtu CAF pro jednotlivé HW konf.

Při porovnání metod založených na dělení do bloků dat je nejrychlejší metodou metoda FMGFD, což je dáno úplnou vektorizací kódu, kdy výpočet je založen pouze na maticových operacích bez použití jakéhokoliv cyklu. Zrychlení doby výpočtu oproti referenční metodě (SMI) je 655x násobná!!! Další výhodou je snadná implementace kódu ze systému Matlab do programovatelných hradlových polí (FPGA), které jsou často využívány při vývoji radarových systémů.

Obdobnou výhodu má i metoda FMCW, kde ale z důvodu nemožnosti (v současné době) vektorizace rozdělení signálu do různých bloků s přesahem dat dochází k nárůstu výpočetní doby. Při analýze jednotlivých řádků kódu bylo zjištěno, že toto rozdělení zabírá cca 70 % celkové výpočetní doby algoritmu. Velice perspektivní je i metoda FFTD, kde její přednosti vyniknou až u výpočtu pomocí GPU (bude uvedena v následující kapitole).

Jednou z mála nevýhod algoritmů FMGFD a FMCW je velká závislost (a tím vyšší náročnost na „správné“ rozdělení vstupních signálů do bloků z hlediska požadovaných vstupních parametrů) rozlišení výsledné CAF na volbě délky bloků vstupních signálů.

7.5.3 Porovnání z hlediska rychlosti výpočtu pro všechny HW konfigurace

Pro celkové vyhodnocení dob výpočtu pro jednotlivé HW metody uvedené v Tab. 30 vyjdeme z jejich rozdělení do uvedených skupin, jež budou samostatně vyhodnoceny. První skupina je tvořena sériovým zpracováním (viz. kapitola 7.5.2), kdy je pro výpočet použito pouze jednojádrového CPU, či je použito pouze jedno jádro z vícejádrového CPU. Tato skupina tvoří referenční hodnoty, na kterých jsou ukázány výhody a případné nevýhody paralelního zpracování.

Druhou skupinu tvoří Multi-core řešení, které využívají pro výpočet již paralelní zpracování dat, ale v rámci pouze jedné PC stanice. Tyto systémy mohou obsahovat jeden či více vícejádrových CPU. V této práci bylo využito pro jednoprosorové (4 jádrové) PC stanice s CPU Intel® Core™ i5-2400 (6MB cache, 3.10 GHz) (kompletní specifikace v kapitole 6.4.2). Z porovnání jednotlivých metod výpočtu CAF je patrné, že u multi-core řešení (HW konfigurace II. – III.) s rostoucím počtem jader klesá doba výpočtu (až na výjimky popsané dále). Bohužel díky nárůstu režijní komunikace mezi jádry, režii pro běh nadstavby pro paralelní zpracování dat, atd. se pokles doby výpočtu ani zdaleka neblíží teoretickým hodnotám, které lze určit pomocí vztahu (6.1). Například u metody SMI dosahovalo urychlení výpočtu oproti referenční hodnotě (zrychlení 1x násobné) hodnot 1,7x (2 jádra), 2,3 (4 jádra) oproti očekávaným hodnotám 2x (2 jádra) a 4x (4 jádra). K urychlení doby výpočtu s rostoucím počtem

jader nedošlo u metody MM a FMGFD. U maticové metody to bylo způsobeno extrémní velikostí matic nutných pro výpočet v této metodě. U metody FMGFD to bylo způsobeno již samo o sobě velice krátkou dobou výpočtu (v řádu 50 ms, cca 667x násobné zrychlení), kde režijní doba již dosahovala úrovně vlastní doby výpočtu (či ji dokonce přesahovala). Z pohledu HW se jedná o perspektivní kompaktní řešení (vše je tvořeno jednou PC stanicí či serverem), které je vhodné zejména pro zpracování úloh, při nichž je požadován velký výpočetní výkon při malém objemu zpracovávaných dat. Nevýhodou tohoto řešení je použití sdílené operační paměti RAM, která je společná pro všechny výpočetní jádra, tj. při například velikosti 4 GB RAM je pro každé jádro k dispozici pouze 1 GB RAM. Pro reálné nasazení u systémů pracujících v reálném čase je nutné uvažovat o velikosti operační paměti pro jedno jádro v řádu 4-8 GB RAM. Výhodou Multi-core řešení je zejména univerzálnost pro téměř libovolný druh výpočtů a velmi vysoký výkon (v případě víceprocesorových systémů).

Třetí rozsáhlá skupina HW konfigurací je založena na využití výpočetního clusteru, do kterého byly vkládány velké naděje pro podstatné urychlení výpočtu (HW konfigurace IV. – IX.). Počet PC stanic v clusteru se pohyboval od 2 PC stanic do 4 PC stanic a počet jader použitých k výpočtu v rozmezí od 2 do 16. Při pohledu na výslednou tabulku hodnot je zřejmé, že nedošlo ke splnění očekávání, doba výpočtu v mnoha případech dokonce s rostoucím počtem jader rostla, což, jak se po podrobné analýze ukázalo, bylo způsobeno neustálým přesunem vstupních dat po síti k jednotlivým výpočetním uzlům, které prováděly vlastní výpočet. Další prodloužení doby výpočtu bylo způsobené náhodnými fluktuacemi rychlosti komunikační datové sítě, kdy se rozdílily mezi dobami výpočtu při shodném HW konfiguraci a shodné metodě výpočtu lišily v řádu desítek procent. Tyto výpadky sítě se objevovaly zcela náhodně a jsou způsobené zřejmě nedokonalostí (nebo spíše nevhodností) Distributed Computing Serveru na operačním systému Windows, kde veškeré výpočty jsou realizované přes SW vrstvu JAVA, která má na operačním systému Windows velké problémy s inicializací tzv. komunikačního socketu. V podstatě lze říci, že doba, než proběhnou veškerá nastavení před začátkem vlastního výpočtu, při kterém dochází ke spuštění celé řady podpůrných procesů v jazyku Java (na kterém je založena paralelizační nadstavba výpočetního systému MATLAB), je tak velká vzhledem k vlastní době výpočtu algoritmu CAF, že nedojde k urychlení výpočtu, ale k jeho zpomalení. Je nutno podotknout, že tento jev lze odstranit využitím platformy Linux, kde k podobným problémům dochází ve výrazně menší míře. Další nevýhodou výpočetního clusteru je jeho poměrně složitá konfigurace a vysoké finanční náklady vzhledem k ostatním HW skupinám uvedeným v této práci.

Čtvrtá a poslední skupina využívá pro výpočet CAF grafické karty s GPU. Jedná se o nejmladší a zároveň neperspektivnější technologii (první verze pro veřejnost byla uveřejněna v roce 2007), která je použita v této práci. Při pohledu na Tab. 30 je zřejmé, že u všech uvedených metod došlo k významnému urychlení, kterým ostatní HW konfigurace u většiny metod nemohou konkurovat, a to i z důvodu zlomkové ceny vzhledem k cenám pracovních Multi-core stanic. Největšího zrychlení bylo dosaženo u metody FMGFD. Metoda byla 667x rychlejší oproti referenční hodnotě. Velkou výhodou využití GPU je možnost použití speciálních akceleračních GPU karet pro numerické výpočty [69], jež dosahují podstatně vyššího výkonu, než dosahuje grafická karta použitá k testování (Gigabyte GTX 580, která je využívána zejména v herním segmentu), a zejména využití pouze pasivního chlazení u grafických karet TESLA. Celkový výpočetní systém pro PCL pak může dosahovat velikosti objemnější PC stanice za cenu, která je podstatně nižší než u všech ostatních HW konfigurací. Jako každá technologie má i tato své nevýhody, které jsou dány především odlišným stylem programování pro GPU a také velikostí operační paměti jednotlivých GPU, avšak tento problém lze vyřešit paralelním zapojením více GPU do jednoho PC systému, kde každá karta bude provádět část výpočtů. Díky tomuto řešení pak dojde k rozdělení náročnosti z hlediska paměti potřebné pro výpočet.

Závěry plynoucí z této práce lze shrnout do několika bodů:

- Z pohledu metody výpočtu CAF patří mezi nejvýhodnější metody (nejrychlejší) metoda FMGFD a FMCW. Pokud lze využít v systému i vyšší faktor decimace (při nekoherentním zpracování), pak i metoda FFTD.
- Z hlediska HW konfigurací se jednoznačně potvrdila výjimečnost výpočtu na GPU, které v budoucnu (v této oblasti) nahradí velké a těžkopádné technologie využívajících clusteru či víceprocesorových systémů.
- Z hlediska univerzálnosti je vhodné zvolit v dnešní době výpočetní systém využívající kombinaci víceprocesorového systému obsahujícího několik GPU karet (záleží na potřebném výkonu).

V hodnocení práce se vyhýbám uvedení konkrétních specifikací HW, a to z důvodu možnosti využití výpočetního systému nejen pro výpočet vzájemné funkce neurčitosti, ale také pro další části systému PCL (adaptivní filtrace, tracking, asociace cílů, atd.). V případě zahrnutí těchto dalších částí do výpočetního systému dojde k nárůstu požadavků na výpočetní výkon.

8 ZÁVĚR

Tato disertační práce se věnuje analýze a optimalizaci (zrychlení) numerického výpočtu vzájemné funkce neurčitosti – CAF, která vyjadřuje korelaci mezi přímým a odraženým signálem obsahujícím informaci o jednom nebo více cílech. CAF tvoří jádro přijímače bistatického pasivního radarového systému či systému pasivní koherentní lokace PCL, který představuje jednu z možností budoucího vývoje radarových systémů. CAF slouží k odhadu parametrů bistatické vzdálenosti a Dopplerových posunů, což jsou stěžejní parametry pro určení následné 3D polohy a rychlosti cílů. Práce obsahuje teoretickou analýzu CAF včetně jejího začlenění do PCL systému, matematický popis CAF s uvedením všech významných vlastností ovlivňujících správnou funkčnost systému. Výpočet CAF je velice numericky (a tím i časově) náročný, což tvoří jednu z nejvýznamnějších překážek pro jeho nasazení ve skutečných PCL systémech pracujících v reálném čase. Zrychlení výpočtu CAF je tedy jedním ze stěžejních problémů pro jejich širší rozvoj. Ke zrychlení výpočtu CAF jsou zvoleny dva základní. První je založen na vhodném výběru hardwaru, na kterém výpočet CAF probíhá. Druhý přístup obsahuje „úpravy“ definičního algoritmu výpočtu pro jejich vyšší optimalizaci. Jako vyhodnocovací kritérium v této práci byla zvolena doba výpočtu CAF s následným uvedením zrychlení doby výpočtu vzhledem k definiční metodě CAF.

8.1 Analýza hardwaru pro výpočet CAF

Z hlediska hardwarového vybavení vhodného pro výpočet CAF došlo k vyhodnocení čtyř základních přístupů. První byl tvořen „běžným“ sériovým zpracováním dat, který sloužil zejména k ověření funkčnosti jednotlivých výpočetních algoritmů CAF. Ostatní přístupy již využívaly paralelní zpracování dat, jež je velice perspektivní z hlediska zrychlení výpočtů.

Z praktického hlediska se jako nejefektivnější, a to napříč všemi metodami výpočtu CAF, ukázal přístup využívající numerický výpočet pomocí grafické karty, tzv. GPU computing, který dosahoval jednoznačně nejkratších výpočetních časů. Jeho nevýhody lze vidět pouze v odlišnosti programovacích technik (týká se všech paralelních metod zpracování), které tento přístup vyžaduje, a zatím omezené velikosti paměti grafických karet, která v současné době dosahuje 8 GB (v porovnání výpočtů pomocí CPU, kde velikost operační paměti RAM může dosahovat téměř neomezené velikosti). Nevýhody tohoto přístupu budou odstraněny s postupným rozšířením paměti GPU.

Dalším analyzovaným přístupem bylo využití tzv. Multi-core systémů s více CPU na jedné základní desce, které lze považovat za jedinou možnou konkurenci GPU systémům především z hlediska jejich univerzálnosti pro libovolný druh výpočtů. V nejbližším časovém horizontu vidím nejvyšší efektivnost ve využití kombinace GPU a Multi-core systémů, a to zejména z důvodu univerzálnosti takového řešení. Při dostatečném rozšíření GPU systémů bude jejich význam nabývat na důležitosti před Multi-core systémy.

Poslední analyzovanou skupinu tvoří využití clusterů, tj. skupiny počítačů propojených sítí, kde každý výpočetní uzel (1 PC) počítá část zpracovávané úlohy. Toto řešení se ukázalo jako velice neefektivní pro výpočet CAF. Hlavním důvodem je režijní přesun dat mezi jednotlivými výpočetními uzly, kde přenos dat určených pro výpočet je časově srovnatelný (či přesahuje) s dobou vlastního výpočtu na uzlu.

8.2 Analýza algoritmů CAF

Výpočet vzájemné funkce neurčitosti dle definičního vztahu je časově velmi neefektivní a pro nasazení do reálných systémů zcela nevyhovující. Tato práce obsahuje soubor algoritmů CAF, založených na různých úpravách definičního vztahu výpočtu, provedené úpravy lze rozdělit do dvou základních skupin. První skupinu tvoří algoritmy, které pracují s celou délkou dat, ve druhé skupině jsou data dělena do jednotlivých bloků. Do první skupiny lze zařadit metody SM, FFT, MM, XC, kde nejkratších dob výpočtu dosahovala metoda FFT, a to zejména z důvodu využití vysoce optimalizovaných algoritmů pro výpočet rychlé Fourierovy transformace. Použitím FFT metody pro sériové zpracování dat bylo v porovnání se sumační metodou SM (referenční) dosaženo téměř 16-ti násobného zrychlení oproti referenčnímu výpočtu. Maticová metoda (MM) výpočtu založená na plně maticové formulaci nenaplnila očekávání, což bylo způsobeno extrémní velikostí matic určených pro zpracování (cca $40 \cdot 10^6$ prvků komplexních čísel). Takové řešení vede, obdobně jako u korelační metody (XC), na extrémní nároky na velikost operační paměti RAM.

Druhou skupinu tvoří algoritmy založené na dělení vstupních dat do bloků. Takové řešení je z hlediska výpočtu výhodnější, a to díky nižší výpočetní náročnosti jednotlivých matematických operací. Tuto skupinu tvoří algoritmy FFTD, FMGFD a FMCW. Metoda FFTD je metoda vycházející z metody FFT a je založena na decimaci signálového součinu, čímž dochází k snižování nároků na výpočet (ovšem současně se snižováním přesnosti výpočtu CAF). Zrychlení oproti referenční metodě je pro faktor decimace 4 přibližně 19-ti násobné. Za maximální faktor decimace lze

považovat hodnotu 10, pro vyšší hodnoty již dochází k velkému poškození fázové informace. Za nejperspektivnější lze považovat metody výpočtu FMCW a FMGFD, kde první ze jmenovaných je založená na výpočtu CAF pomocí částečných korelací a dosahovala téměř 45-ti násobného zrychlení výpočtu oproti referenční hodnotě. Vyšší optimalizaci této metody brání paralelizace vlastního rozdělení do bloků s přesahem dat, která tvoří cca 70 % celkové výpočetní doby algoritmu. Poslední analyzovanou metodou je metoda FMGFD, která je založená na výpočtu CAF ve frekvenční oblasti. Díky kompletní vektorizaci kódu, tj. použití pouze maticových operací, došlo k téměř 667-ti násobnému zrychlení výpočtu proti referenční hodnotě.

Při analýze dob výpočtu jednotlivých metod CAF pro všechny hardwarové konfigurace lze spatřit „kopírování“ výsledků mezi sériovým a paralelním zpracováním. Metody, které dosahovaly nejlepších výsledků při sériovém zpracování, dosahovaly nejlepších výsledků i při paralelním zpracování (zejména při využití GPU). Nejvyššího zrychlení dosáhla metoda FMGFD na hardwarové konfiguraci využívající pro výpočet GPU.

Z pohledu výpočtu CAF jsou nejvýhodnější metody FMGFD, FMCW a FFTD při využití GPU zpracování. V současné době je z hlediska univerzality vhodné pro výpočet CAF i využití Multi-core systémů.

8.3 Možnosti další práce

Rozvoj číslicového zpracování signálů, vývoj algoritmů či rozvoj hardwarových prostředků tvoří dynamicky se rozvíjející odvětví, kde s každou další generací hardwarového vybavení dochází k nárůstu výpočetního výkonu, čímž aktuálnost této práce je poplatná době, ve které je napsána. Možné pokračování této práce vidím zejména v aplikaci nejvýhodnějších algoritmů CAF do reálného HW zařízení, které je založené na digitálních signálových procesorech či hradlových polích. Další možné výzkumné práce vidím ve využití Multi-GPU zařízení (zařízení obsahující více GPU karet pro výpočet), na kterém by nedocházelo pouze k výpočtu CAF, ale také ke zpracování ostatních částí PCL systému jako jsou adaptivní filtrace, tracking, asociace cílů, atd.

LITERATURA

- [1] WILLIS, N. J., Bistatic Radar. 2nd ed. Raleigh: SciTech Pub, 1995. ISBN 18-911-2145-6.
- [2] BLAKE, Lamont V. Radar range-performance analysis. Norwood, MA: Artech House, 1986, 443 s. ISBN 08-900-6224-2.
- [3] IEEE standard radar definitions. New York: Institute of Electrical and Electronics Engineers, 2008. ISBN 978-073-8157-450.
- [4] SKOLNIK, M. I. Introduction to radar systems. 2d ed. New York: McGraw-Hill, 1980, 581 s. ISBN 00-705-7909-1.
- [5] NATHANSON, F. E. Radar design principles signal processing and the environment. 2nd ed. Mendham, N.J: SciTech Pub, 1999. ISBN 18-911-2109-X.
- [6] BEZOUŠEK, P., ŠEDIVÝ, P. Radarová technika. Vyd. 1. Praha: Vydavatelství ČVUT, 2004. ISBN 978-800-1030-363.
- [7] HRDINA, Z. Rádiové určování polohy: Družicový systém GPS. 1. vyd. Praha: Vydavatelství ČVUT, 1999, 259 s. ISBN 80-010-1386-3.
- [8] BARTON, D. K. a S. LEONOV. Radar technology encyclopedia. Boston: Artech House, 1997, 511 s. ISBN 08-900-6893-3.
- [9] WILLIS, N. J. a H. GRIFFITHS. Advances in bistatic radar. Raleigh, NC: SciTech Pub., 2007, 493 s. ISBN 18-911-2156-1.
- [10] HAYKIN, S. Adaptive filter theory. 4th ed. Upper Saddle River: Prentice Hall, 2002, 920 s. ISBN 01-309-0126-1.
- [11] SAYED, A. H. Fundamentals of adaptive filtering. New York: IEEE Press, 2003, 1125 s. ISBN 04-714-6126-1.
- [12] Passive radar [online]. December 2007 [cit. 2012-03-10]. Dostupné z: http://en.wikipedia.org/wiki/Passive_radar
- [13] SCHOENENBERGER, J.G. a J.R. FORREST. Principles of independent receivers for use with co-operative radar transmitters. Radio and Electronic Engineer. 1982, roč. 52, č. 2, s. 93-101. ISSN 00337722. DOI: 10.1049/ree.1982.0013.
- [14] THOMPSON, E. C. Bistatic radar noncooperative illumination synchronization techniques. In: Radar Conference, 1989., Proceedings of the 1989 IEEE National. Santa Monica, CA, USA, 1989, s. 29-34.

- [15] HOWLAND, P.E., MAKSIMIUK, D., REITSMA, G. FM radio based bistatic radar. IEE Proceedings - Radar, Sonar and Navigation. 2005, roč. 152, č. 3, s. 107-. ISSN 13502395.
- [16] OLSEN, K.E. a K. WOODBRIDGE. FM based passive bistatic radar range resolution improvement. In: Radar Symposium (IRS), 2011 Proceedings International. Kjeller, Norway, 2011, s. 327-332.
- [17] OLSEN, K.E., WOODBRIDGE, K. ANDERSEN, I.A. FM based passive bistatic radar range resolution improvement Part II. In: Radar Symposium (IRS), 2010 Proceedings International. Vilnius, Lithuania, 2010, s. 1 - 8.
- [18] COLEMAN, C., YARDLEY, H. Passive bistatic radar based on target illuminations by digital audio broadcasting. IET Radar, Sonar. 2008, roč. 2, č. 5, s. 366-375. ISSN 17518784.
- [19] POUILLIN, D. Passive detection using digital broadcasters (DAB, DVB) with COFDM modulation. IEE Proceedings - Radar, Sonar and Navigation. 2005, roč. 152, č. 3, s. 143-. ISSN 13502395.
- [20] YARDLEY, H. J. Bistatic Radar Based on DAB Illuminators: The Evolution of a Practical System. In: Radar Conference, 2007 IEEE. Univ. of Adelaide, Adelaide, USA, 2007, s. 688-692. ISBN 1-4244-0284-0.
- [21] GRIFFITHS, H. D. a N. R. W. LONG. Television-based bistatic radar. IEE proceedings. F, Communications, radar, and signal processing. 1986, roč. 7, č. 177, s. 649-657. ISSN 0143-7070.
- [22] HOWLAND, P. F. Target tracking using television-based bistatic radar. IEE Proceedings - Radar, Sonar and Navigation. 1999, roč. 146, č. 3, s. 166-174. ISSN 13502395.
- [23] WANG, H., WANG, J., ZHONG, L. Mismatched filter for analogue TV-based passive bistatic radar. IET Radar, Sonar. 2011, roč. 5, č. 5, s. 573-581. ISSN 17518784.
- [24] SAINI, R., CHERNIAKOV, M. DTV signal ambiguity function analysis for radar application. IEE Proceedings - Radar, Sonar and Navigation. 2005, roč. 152, č. 3, s. 133-. ISSN 13502395.
- [25] CHRISTIANSEN, J. M. a K. E. OLSEN. Range and Doppler walk in DVB-T based Passive Bistatic Radar. In: 2010 IEEE Radar Conference: Crystal Gateway Marriott, Washington DC, USA, May 10-14, 2010: proceedings: Radar 2010, global innovation in radar. Piscataway, N.J.: IEEE, 2010, 620 - 626. ISBN 978-1-4244-5811-0.

- [26] OUT, J. Sea target detection using passive DVB-T based radar. In: 2008 International Conference on Radar. Adelaide, USA: IEEE, 2009, 695 - 700. ISBN 978-1-4244-2321-7.
- [27] TAN, D.K.P., SUN, H., LU, Y., LESTURGIE, M. CHAN, H.L. Passive radar using Global System for Mobile communication signal: theory, implementation and measurements. IEE Proceedings - Radar, Sonar and Navigation. 2005, roč. 152, č. 3, s. 116- 123. ISSN 13502395.
- [28] TAN, D.K.P., SUN, H., LU, Y., LIU, W. Feasibility analysis of GSM signal for passive radar. In: Proceedings of the 2003 IEEE Radar Conference: May 5 - 8, 2003, Marriott Hotel, Huntsville, Alabama. Piscataway, NJ: IEEE Operations Center, 2003, s. 425 - 430. ISBN 0-7803-7920-9.
- [29] TAN, D.K.P., SUN, H., LIU, W. Design and implementation of an experimental GSM based passive radar. In: 2003 proceedings of the International Conference on Radar: Adelaide, Australia, September 3-5, 2003. Piscataway, N.J.: IEEE, 2003, s. 418-422. ISBN 0780378709.
- [30] NEYT, X., RAOUT, J., KUBICA, M. a V., ROGUES, S. Feasibility of STAP for passive GSM-based radar. In: 2006 IEEE International Radar Conference record: April 24-27 2006, Verona, New York, USA. Institute of Electrical and Electronics Engineers, 2006, s. 6. ISBN 0-7803-9496-8.
- [31] Vysílací služby. České Radiokomunikace a.s. [online]. [cit. 2012-03-10]. Dostupné z: <http://www.radiokomunikace.cz/vysilaci-sluzby.html>
- [32] LYNCH, D. Introduction to RF stealth. Raleigh, NC: SciTech Pub., 2004, 575 s. ISBN 08-634-1349-8.
- [33] Stealth technology. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-10]. Dostupné z: http://en.wikipedia.org/wiki/Stealth_technology
- [34] GRANT, R. The Radar Game: Understanding Stealth and Aircraft Survivability. Virginia, USA: IRIS Independent Research, 1998. ISBN 703-875-8745.
- [35] PROAKIS, J. G., MANOLAKIS, D. G. Digital signal processing. 4th ed. Upper Saddle River: Pearson Prentice Hall, 2007, 1084 s. ISBN 01-318-7374-1.
- [36] TEARNS, S. D. Digital signal processing with examples in MATLAB. Boca Raton: CRC Press, 2003, 336 s. ISBN 08-493-1091-1.

- [37] Oppenheim, A. V., Schafer, R. W. Discrete-time signal processing. 3rd ed., International ed. Upper Saddle River, N.J: Pearson Education, 2009. ISBN 01-320-6709-9.
- [38] Cherniakov, M. Bistatic radar: principles and practice. Chichester, England: John Wiley, 2007. ISBN 978-047-0026-304.
- [39] BEZOUŠEK, P. Signály a soustavy. Univerzita Pardubice, Fakulta elektrotechniky a informatiky, 2011. Přednášky.
- [40] STEIN, S. Algorithms for ambiguity function processing. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1981, roč. 29, č. 3, s. 588-599. ISSN 0096-3518.
- [41] TOLIMIERI, R., WINOGRAD, S. Computing the ambiguity surface. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1985, roč. 33, č. 5, s. 1239-1245. ISSN 0096-3518.
- [42] AUSLANDER, L., TOLIMIERI, R. Computing decimated finite cross-ambiguity functions. IEEE Transactions on Acoustics, Speech, and Signal Processing. roč. 36, č. 3, s. 359-364. ISSN 00963518.
- [43] COOLEY J.W., WINOGRAD, S. A limited range discrete Fourier transform algorithm. In: Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '80. Yorktown Heights, N.Y., USA, 1980, s. 213-217.
- [44] PORAT, B. A course in digital signal processing. New York: John Wiley, 1997, 602 s. ISBN 04-711-4961-6.
- [45] Brown, W. A., Loomis, H. H., Digital Implementations of Spectral Correlation Analyzers, IEEE Transactions on Signal Processing, vol. 41, no. 2, pp. 703-720, Feb. 2003.
- [46] Convolution. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-10]. Dostupné z: <http://en.wikipedia.org/wiki/Convolution>
- [47] History of computer clusters. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2012-03-10]. Dostupné z: http://en.wikipedia.org/wiki/History_of_computer_clusters
- [48] History of supercomputing. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-03-10]. Dostupné z: http://en.wikipedia.org/wiki/History_of_supercomputing

- [49] Computer cluster. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-03-10]. Dostupné z: [http://en.wikipedia.org/wiki/Cluster_\(computing\)](http://en.wikipedia.org/wiki/Cluster_(computing))
- [50] Mathworks. [online]. [cit. 2012-03-10]. Dostupné z: <http://www.mathworks.com>
- [51] Parallel computing. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-03-10]. Dostupné z: http://en.wikipedia.org/wiki/Parallel_computing
- [52] Parallel computing. [online]. 2010 [cit. 2012-03-10]. Dostupné z: http://www.dmoz.org/Computers/Parallel_Computing/
- [53] LIN, Yun Calvin, SNYDER, L. Principles of parallel programming. Pearson internat. ed. Boston, Pearson Addison Wesley, 2009, 338 s. ISBN 03-215-4942-2.
- [54] KEPNER, J. Parallel MATLAB for multicore and multinode computers. Philadelphia: Society for Industrial and Applied Mathematics, 2009, 253 s. ISBN 978-0-898716-73-3.
- [55] MATHWORKS. Náповěda systému Matlab. 2012.
- [56] BANERJEE, U. Loop transformations for restructuring compilers: the foundations. Boston: Kluwer Academic, 1993, 305 s. ISBN 07-923-9318-X.
- [57] BANERJEE, U. Loop transformations for restructuring compilers: Loop paralelization. Boston: Kluwer Academic, 1993, 305 s. ISBN 07-923-9318-X.
- [58] BANERJEE, U. Loop transformations for restructuring compilers: Dependence Analysis. Boston: Kluwer Academic, 1993, 305 s. ISBN 07-923-9318-X.
- [59] GPU Computing. Nvidia [online]. [cit. 2012-03-10]. Dostupné z: http://www.nvidia.com/object/GPU_Computing.html
- [60] GPGPU. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-03-10]. Dostupné z: <http://en.wikipedia.org/wiki/GPGPU>
- [61] SANDERS, J. CUDA by example: an introduction to general-purpose GPU programming. 1st print. Upper Saddle River: Addison-Wesley, 2010, 290 s. ISBN 978-0-13-138768-3.
- [62] Matlab Parallel Computing Toolbox. Mathworks [online]. 2012 [cit. 2012-03-10]. Dostupné z: <http://www.mathworks.com/products/parallel-computing/>

- [63] Matlab Distributed Computing Server. Mathworks [online]. 2012 [cit. 2012-03-10]. Dostupné z: <http://www.mathworks.com/products/distriben/>
- [64] AccelerEyes Jacket. AccelerEyes [online]. 2012 [cit. 2012-03-10]. Dostupné z: <http://www.accelereyes.com>
- [65] Overhead (Computing). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-03-10]. Dostupné z: http://en.wikipedia.org/wiki/Computational_overhead
- [66] CHERNIAKOV, M. Bistatic radar: emerging technology. Hoboken, NJ: J. Wiley, 2008. ISBN 9780470026311.
- [67] Improving Performance with Parallel Computing. Mathworks [online]. 2012 [cit. 2012-03-10]. Dostupné z: <http://www.mathworks.com/help/toolbox/optim/ug/briutum.html>
- [68] Parallel Computing in Optimization Toolbox Functions. Mathworks [online]. 2012 [cit. 2012-03-10]. Dostupné z: <http://www.mathworks.com/help/toolbox/optim/ug/briutqn-1.html>
- [69] Tesla Workstation Solutions. Nvidia [online]. 2012 [cit. 2012-03-10]. Dostupné z: http://www.nvidia.co.uk/page/personal_computing.html

SEZNAM OBRÁZKŮ

Obr. 1	Bistatický trojúhelník.....	7
Obr. 2	Elipsoid možných výskytů cílů	8
Obr. 3	Ukázka rozmístění PCL systému	10
Obr. 4	Blokové schéma přijímače pasivního radaru	12
Obr. 5	Spektrum vstupního signálu	17
Obr. 6	Spektrum pomocného signálu $\tilde{S}(\omega)$	18
Obr. 7	Spektrum analytického signálu	19
Obr. 8	Spektrum komplexní obálky signálu	20
Obr. 9	Ukázka vzájemné funkce neurčitosti	28
Obr. 10	Ukázka špatného rozlišení dvou cílů	31
Obr. 11	Vliv šířky pásma na šířku hlavního laloku $ CAF $	31
Obr. 12	Řez CAF v maximu při konstantním časového posuvu $\tau = \tau_{D0}$	33
Obr. 13	Schematické zobrazení vazeb mezi jednotlivými metodami.....	37
Obr. 14	Grafické zobrazení výpočtu CAF pomocí metody FFT	41
Obr. 15	Grafické zobrazení rozdělení vzorků do m skupin.....	45
Obr. 16	Grafické zobrazení výpočtu CAF pomocí metody FFT s decimací signálového součinu	46
Obr. 17	Grafické zobrazení výpočtu CAF pomocí metody ve frekvenční oblasti	49
Obr. 18	Princip metody výpočtu CAF založený na výpočtu částečných korelací....	51
Obr. 19	Principiální zapojení vícejádrových procesorů.....	56
Obr. 20	Principiální zapojení víceprocesorových systémů	56
Obr. 21	Principiální zapojení clusteru [49].....	57
Obr. 22	Porovnání CPU a GPU z hlediska počtu jader [59]	58

SEZNAM TABULEK

Tab. 1.	Seznam vysílačů pro různé služby v ČR [31]	11
Tab. 2.	Algoritmus výpočtu CAF sumační metodou (SM).....	38
Tab. 3.	Algoritmus výpočtu CAF korelační metodou (XC)	40
Tab. 4.	Algoritmus výpočtu CAF pomocí metody (FFT)	41
Tab. 5.	Algoritmus výpočtu CAF maticovou metodou (MM)	44
Tab. 6.	Algoritmus výpočtu CAF pomocí metody FFTD	47
Tab. 7.	Algoritmus výpočtu CAF ve frekvenční oblasti (FMGFD)	50
Tab. 8.	Algoritmus výpočtu CAF pomocí metody částečných korelací (FMCW)	53
Tab. 9.	Seznam HW konfigurací.....	63
Tab. 10.	Seznam testovaných metod výpočtu CAF.....	64
Tab. 11.	Seznam vstupních parametrů	66
Tab. 12.	Naměřené doby výpočtu CAF pomocí sumační metody I.	68
Tab. 13.	Vyhodnocení Sumační metody I. pro všechny HW konfigurace	69
Tab. 14.	Vyhodnocení Sumační metody pro všechny HW konfigurace.....	71
Tab. 15.	Vyhodnocení Sumační metody II. pro všechny HW konfigurace	72
Tab. 16.	Vyhodnocení Korelační metody pro všechny HW konfigurace	74
Tab. 17.	Souhrnné vyhodnocení korelační metody pro všechny HW konfigurace .	75
Tab. 18.	Vyhodnocení FFT metody pro všechny HW konfigurace	76
Tab. 19.	Souhrnné vyhodnocení FFT metody pro všechny HW konfigurace	77
Tab. 20.	Vyhodnocení maticové metody pro všechny HW konfigurace.....	79
Tab. 21.	Souhrnné vyhodnocení FMCW metody pro všechny HW konfigurace	80
Tab. 22.	Vyhodnocení FFTD metody pro všechny HW konfigurace.....	81
Tab. 23.	Souhrnné vyhodnocení FFTD metody pro všechny HW konfigurace	82
Tab. 24.	Vyhodnocení FMGFD metody pro všechny HW konfigurace	83
Tab. 25.	Souhrnné vyhodnocení FMGFD metody pro všechny HW konfigurace ...	84
Tab. 26.	Vyhodnocení FMCW metody pro všechny HW konfigurace.....	85
Tab. 27.	Souhrnné vyhodnocení FMCW metody pro všechny HW konfigurace	86
Tab. 28.	Seznam rozdílových povrchů	87

Tab. 29. Shrnutí vybraných parametrů u jednotlivých algoritmů CAF.....	89
Tab. 30. Celkové vyhodnocení metod výpočtu CAF pro jednotlivé HW konf.....	90

SEZNAM PUBLIKACÍ

- [1] SCHEJBAL, V., D. ČERMÁK, Z. NĚMEC, J. PIDANIČ, J. KONEČNÝ, P. BEZOUŠEK a O. FIŠER. Multipath Propagation of UWB through-Wall Radar and EMC Phenomena. Radioengineering. Praha: Fakulta elektrotechnická ČVUT - Středisko vědeckotechnických informací, 2006, roč. 15, č. 4, s. 6. ISSN 1210-2512.
- [2] PIDANIČ, J. Detekce vozidel na přejezdu pomocí UWB radarů. Sdělovací technika: telekomunikace - elektronika - multimédia. Praha: Petr Beneš v nakladatelství Sdělovací technika s. r. o, 2007, č. 13, s. 3. ISSN 0036-9942.
- [3] PIDANIČ, J. UWB radary. Sdělovací technika: telekomunikace - elektronika - multimédia. Praha: Petr Beneš v nakladatelství Sdělovací technika s. r. o, 2007, č. 13, s. 3. ISSN 0036-9942.
- [4] PIDANIČ, J., D. ČERMÁK a V. SCHEJBAL. Through-wall Propagation Measurements. In: BONEFAČIĆ, Davor. ICECom 2007: 19th International Conference on Applied Electromagnetics and Communications, 24-26 September 2007, Dubrovnik, Croatia. Zagreb, Croatia: KoREMA, 2007, s. 4. ISBN 978-953-6037-50-6.
- [5] PIDANIČ, J., D. ČERMÁK a V. SCHEJBAL. Through-wall Propagation Measurements. In: 2007 17th International Conference Radioelektronika: Brno, Czech Republic, 24-25 April 2007. Piscataway, NJ: IEEE, 2007, s. 4. ISBN 1-4244-0821-0.
- [6] KRÍŽ, J., V. KRČMÁŘ, J. PIDANIČ a V. SCHEJBAL. Control of antenna beam width. In: Proceedings of the 14th Conference on Microwave Techniques: COMITE 2008 ; April 23-24, 2008, Prague. Piscataway, NJ: IEEE Service Center, 2008, s. 7. ISBN 978-1-4244-2137-4.
- [7] PIDANIČ, J., D. ČERMÁK, SCHEJBAL, V. Gain Estimation of Doubly Curved Reflector Antenna. Radioengineering. Praha: Fakulta elektrotechnická ČVUT - Středisko vědeckotechnických informací, 2008, roč. 3, č. 3, s. 4. ISSN 1210-2512.
- [8] SCHEJBAL, V., J. PIDANIC, V. KOVARIK a D. CERMAK. Accuracy Analyses of Synthesized-Reference-Wave Holography for Determining Antenna Radiation Characteristics. IEEE Antennas and Propagation Magazine. 2008, roč. 50, č. 6, s. 89-98. ISSN 1045-9243. DOI: 10.1109/MAP.2008.4768928.

- [9] BEZOUŠEK, P., J. PIDANIČ a M. HÁJEK. Analýza metody asociace dat v multistatickém radarovém systému. Slaboproudý obzor. 2009, roč. 64, č. 5, s. 8. ISSN 0037-668X.
- [10] PIDANIČ, J., V. SCHEJBAL a D. ČERMÁK. Analysis of Periodic Errors for Synthesized-Reference-Wave Holography. Radioengineering. Praha: Fakulta elektrotechnická ČVUT - Středisko vědeckotechnických informací, 2009, roč. 18, č. 4, s. 5. ISSN 1210-2512.
- [11] SCHEJBAL, V., D. CERMAK, J. PIDANIC a O. FISER. Diffraction over an Earth – Comparison of Methods. In: Proceedings of 15th Conference on Microwave Technique. Brno, 2010, s. 4. ISBN 978-1-4244-6341-1.
- [12] KRIZ, Josef, Vitezslav KRCMER, Jan PIDANIC a Vladimír SCHEJBAL. Antenna Beamwidth Control [Antenna Designer's Notebook. IEEE Antennas and Propagation Magazine. 2010, roč. 52, č. 1, s. 163-170. ISSN 1045-9243.
- [13] PIDANIČ, J. a V. SCHEJBAL. Broadband Approximations for Doubly Curved Reflector Antenna. Radioengineering. Praha: Fakulta elektrotechnická ČVUT - Středisko vědeckotechnických informací, 2010, roč. 19, č. 4, s. 5. ISSN 1210-2512.
- [14] SCHEJBAL, V., O. FISER a J. PIDANIC. Propagation Over Terrain - Comparison of Methods. In: Antennas and Propagation (EUCAP), Proceedings of the 5th European Conference. Rome, Italy, 2011, s. 4. ISBN 978-1-4577-0250-1.
- [15] SCHEJBAL, V., O. FISER a J. PIDANIC. Comparisons of Approximate and Exact Solutions for Forward Scattering. In: Antennas and Propagation (EUCAP), Proceedings of the 5th European Conference. Rome, Italy, 2011, s. 4. ISBN 978-1-4577-0250-1.
- [16] SCHEJBAL, V., J. PIDANIC a O. FISER. Broadband Approximation of Radiation Patterns for Doubly Curved Reflector Antennas. IEEE antennas. 2011, č. 53, s. 7. ISSN 1045-9243.
- [17] BEZOUŠEK, P. - PIDANIČ, J. - POLA, M. Detection of Multiple Targets by a Multistatic Radar. Perner's Contacts, 2009, vol. 4, no. 4, s. 6-20. ISSN: 1801-674X.
- [18] PIDANIČ, J. Parallel Computing OF Cross ambiguity function with brute-force methods. Perner's Contacts, 2011, vol. 6, no. 5, s. 6-20. ISSN: 1801-674X.