

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická

Zpracování dat z FMCW automobilového radaru z městského prostředí

Jan Šmolcnp

Vedoucí: Ing. Viktor Adler, Ph.D.
Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šmolcnop** Jméno: **Jan** Osobní číslo: **492066**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra elektromagnetického pole**
Studijní program: **Elektronika a komunikace**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Zpracování dat z FMCW automobilového radaru z městského prostředí

Název bakalářské práce anglicky:

Automotive FMCW radar's data processing from urban environment

Pokyny pro vypracování:

Vypracujte rešerši principu činnosti MIMO FMCW radarů. Popište, jak určit vzdálenost, rychlost, elevaci a azimut radarových cílů. Analyzujte dostupná data ze čtyř radarů a kamery z jízdy automobilu po městském prostředí. Ze změřených fázorů ze všech elementů virtuálního anténního pole radarů zjistěte pozici radarových cílů pomocí dostupných kalibračních dat. Detekované cíle vhodným výpočetním nástrojem zobrazte v souřadnicovém systému vozidla a pokuste se je identifikovat i v dostupných videozáznamech.

Seznam doporučené literatury:

- [1] Brooker, M., G.: Understanding Millimetre Wave FMCW Radars, Proceedings of 1st International Conference on Sensing Technology, New Zealand, 2005
- [2] Winkler, V.: 'Range Doppler detection for automotive FMCW radars,' 2007 European Radar Conference, 2007
- [3] Sandeep, R.: MIMO Radar, Application Report, Texas Instruments, 2018
- [4] Zhizhang Chen, Gopal Gokeda, Yiqiang Yu, Introduction to Direction-of-Arrival Estimation, Artech House, 2010
- [5] Li, J., Stoica, P.: MIMO Radar Signal Processing, John Wiley & Sons, Inc., New York, 2009
- [6] Bezoušek, P., Šedivý, P.: Radarová technika, Vydavatelství ČVUT, Praha, 2004

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Viktor Adler, Ph.D. katedra elektromagnetického pole FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **03.02.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Viktor Adler, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

Děkuji Ing. Viktoru Adlerovi, Ph.D. a Ing. Milanu Kvičerovi, Ph.D. za jejich ochotu a trpělivost při odpovídání na mé dotazy při psaní této práce.

Prohlášení

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v seznamu literatury.

V Praze dne 20. 5. 2022

Abstrakt

Tato práce se zabývá popisem automobilového FMCW radaru a jeho základními principy činnosti. Cílem práce je shrnout jak radary tohoto typu fungují, jakým způsobem pracují s přijatým signálem a následně v prostředí MathWorks MATLAB naprogramovat zpracování změřených fázorů ze všech elementů virtuálního anténního pole. K analýze byla využita data poskytnuta společností Valeo autoklimatizace, k.s.

Klíčová slova: FMCW, analýza signálu, automobilový radar, azimut, elevace, MATLAB, FFT

Vedoucí: Ing. Viktor Adler, Ph.D.

Abstract

This thesis focuses on the description of automotive FMCW radar. The goal of the thesis is to summarize basic principles of these radars, the signal processing and to write a script in the environment MathWorks MATLAB to analyze the sampled phasors from all elements of virtual antenna array. The analyzed data were provided by the company Valeo autoklimatizace, k.s.

Keywords: FMCW, signal processing, automotive radar, azimuth, elevation, MATLAB, FFT

Obsah

1 Úvod	1
2 Teoretická část	3
2.1 Komponenty radaru	3
2.2 Chirp	3
2.3 Princip FMCW radaru	5
2.3.1 Určení vzdálenosti	7
2.3.2 Určení rychlosti	9
2.3.3 Určení azimutu	10
2.3.4 Určení elevace	10
2.4 Princip MIMO	11
2.4.1 Multiplex	12
2.5 Efektivní odrazná plocha	12
2.5.1 Koutové odražeče	13
2.6 Metody zpracování dat	14
2.6.1 2D FFT	14
2.6.2 Super resolution metody	15
3 Praktická část	17
3.1 Charakteristika dat a testovací jízdy	17
3.1.1 Rozmístění radarů	17
3.1.2 Anténní pole	18
3.1.3 Struktura dat	18
3.1.4 Kalibrační křivky	20
3.2 Zpracování dat	21
3.2.1 Příprava a filtrace	22
3.2.2 FoV radarů	23
3.2.3 Výpočetní cyklus	24
3.2.4 Vykreslení dat do 3D grafu ..	30
3.2.5 Dosažené výsledky	32
4 Závěr	37
Seznam zkratk	39
Literatura	41
Přílohy	43

Obrázky

2.1 Blokové schéma FMCW radaru.	4
2.2 Chirp v časové oblasti pro dvě periody.	4
2.3 FMCW signál pila pro čtyři period.	5
2.4 Určení azimutu s použitím dvou antén [6].	10
2.5 Princip MIMO [6].	11
2.6 Demonstrace anténního pole bez principu MIMO [6].	11
2.7 Dvoudimenzionální virtuální anténní pole [6].	12
2.8 Odraz paprsku v koutovém odražeči.	15
2.9 2D FFT [5].	15
2.10 Range-Doppler plot.	16
3.1 Rozložení radarů na vozidle (Zdroj Valeo autoklimatizace, k.s.).	18
3.2 Ukázka dodaných dat před zpracováním.	19
3.3 Kalibrační křivka pro kombinaci antén TX0_RX1.	21
3.4 Kalibrační křivka pro všechny kombinace antén.	22
3.5 Ukázka filtrace dat.	23
3.6 Maticе FoV radaru 3. Žlutá barva reprezentuje zorné pole, kde jsou zjištěné cíle validní.	23
3.7 Ukázka kódu interpolace FoV (Field of view) matic pro jednotlivé radary.	24
3.8 Převod komplexních vzorků signálu na algebraický tvar a následné určení fáze.	25
3.9 Násobení komplexního čísla s fází přijatého signálu s komplexním číslem s fází z kalibrační křivky.	25
3.10 Ukázka násobení dvou komplexních čísel v závislosti na vzájemném fázovém posunu.	26
3.11 Ukázka jak ovlivní rozdíl fáze komplexních čísel absolutní hodnotu jejich součtu.	27
3.12 Nalezení maxima a převod indexu matice na azimut a elevaci.	27
3.13 Lokální souřadnice vůči radaru. Pohled na rovinu X-Y ze směru Z.	28
3.14 Lokální souřadnice Z vůči radaru.	29
3.15 Transformace lokálních souřadnic v kódu.	30
3.16 3D graf zobrazující detekované cíle.	31
3.17 Náhled z kabiny vozidla na vnější prostředí. Obrázek je vyjmutý z dodaného videa z testovací jízdy.	31
3.18 Ukázka podezřelých cílů v 3D grafu. V obrázku nejsou zobrazeny informace o souřadnicích, rychlosti, azimutu, atd., protože pro identifikaci vadných dat nejsou podstatné.	33
3.19 Vadná data způsobující chybné výsledky.	33
3.20 Cíle, jejichž relativní vzdálenost od cílů podle Valeo autoklimatizace, k.s. je větší než 20 % z pohledu na rovinu XZ.	34
3.21 Ukázka dat, která mají zaměněné pořadí řádků.	35
3.22 Histogram rozdílů azimutů vypočtených a určených podle Valeo autoklimatizace, k.s.	35
3.23 Příprava kreslicí plochy grafu a cyklus postupného vykreslení dat.	36

Tabulky

2.1 Tabulka RCS v mikrovlnném pásmu pro různé objekty [10].	14
2.2 Hodnoty efektivní odrazné plochy člověka na různých kmitočtech mikrovlnného pásma [10].	14
3.1 Rozmístění radarů na vozidle v souřadnicích vůči středu zadní osy, posun v azimutu a označení jednotlivých radarů.	18

Kapitola 1

Úvod

Dnešní automobily jsou vybaveny velkým množstvím elektronických systémů, které pomáhají udržovat bezpečnost na silnicích (např. Lane Assist - udržuje automobil v jízdním pruhu, Adaptive cruise control - automaticky udržuje bezpečnou vzdálenost od vozidla jedoucího vpředu nebo BLIS - detekuje vozidla v takzvaném mrtvém úhlu). Dalším krokem ke zvýšení bezpečnosti je autonomní řízení vozidel. Jedná se o systém, schopný pomocí velkého množství senzorů detekovat překážky, semaforey nebo dopravní značení a na základě toho se navigovat v prostoru. Hojně využívané senzory k tomuto účelu jsou FMCW (Frequency modulated continuous wave) radary [1]. Tyto radary mají velkou výhodu, že nejsou do takové míry ovlivněny povětrnostními nebo světelnými podmínkami jako ostatní senzory a jsou schopné měřit vzdálenost, rychlost, azimut a také elevaci cíle. Mohou být využity ke sledování blízkého okolí vozidla (detekce obrubníků nebo okolních překážek při parkování), ale i na dlouhé vzdálenosti (např. při jízdě na dálnici ke sledování rychlosti automobilu před vozidlem).

Tato práce si klade za cíl zpracovat řešerši principu MIMO (Multiple input multiple output) FMCW radarů. Jakým způsobem lze určit vzdálenost a rychlost cíle a také azimut a elevaci vůči radaru. V teoretické části je popsáno, jakým způsobem je určena vzdálenost a rychlost cíle z frekvence přijatého signálu nebo jak se využívá principu MIMO ke zvýšení přesnosti měření azimutu a elevace. V praktické části je cílem analyzovat data získaná ze společnosti Valeo autoklimatizace, k.s. z testovací jízdy ze čtyř radarů a demonstrovat jakým způsobem se provádí určování azimutu a elevace v praxi u reálných radarů.

Kapitola 2

Teoretická část

Tato kapitola se věnuje teoretickému popisu problematiky. V textu jsou ponechána některá anglická slova bez českého překladu, jelikož jejich ekvivalent buď neexistuje, nebo je nepřesný.

2.1 Komponenty radaru

FMCW radar se skládá z několika základních bloků, které jsou znázorněny na obrázku 2.1. Prvním je napětím řízený oscilátor (VCO), který generuje chirp (sekce 2.2). Pro vysílání signálu se v minulosti hojně využívalo pásmo na frekvenci 24 GHz, ovšem v dnešní době se přechází na pásmo 77 GHz nebo 79 GHz. Tento přechod je zejména způsoben legislativou než použitím odlišné technologie [2].

Poté následuje dělič výkonu (PD), který signál rozdělí do dvou větví. Je důležité aby byl správně impedančně přizpůsoben a neměl příliš vysoký útlum.

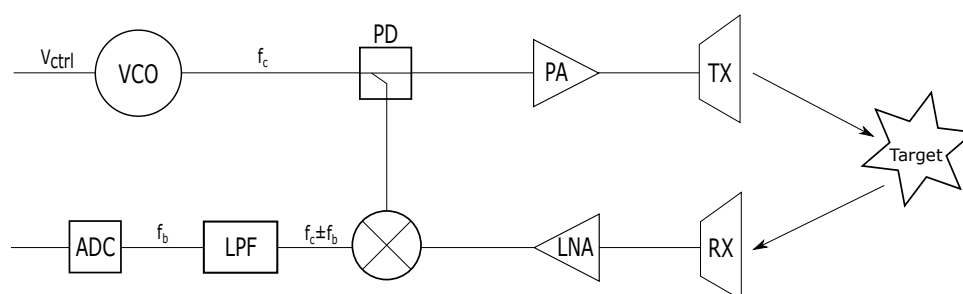
Jedna větev pokračuje skrz zesilovač (PA) do vysílací antény (TX), kde se vyzáří do prostoru. Druhá větev je spojena se směšovačem. Z přijímací antény (RX) putuje signál skrz nízkošumový zesilovač (LNA), u kterého je podstatné, aby do signálu nevnášel další šum, jelikož by poté nemuselo být možné ze signálu filtrací a další analýzou získat potřebnou informaci.

Následuje již zmiňovaný směšovač, který vytvoří mezifrekvenční signál. Ten je dále upraven filtrem dolní propust (LPF), kde se odstraní vysokofrekvenční rušení a také nepotřebná část signálu vytvořená směšovačem.

Posledním prvkem ve schématu je analogově digitální převodník (ADC), který vzorkuje signál a připravuje ho pro digitální zpracování. Rozhodujícím parametrem je vzorkovací frekvence, která se typicky pohybuje v desítkách megahertzů.

2.2 Chirp

Vlny jsou vysílány v tzv. *chirpech*. Jedná se o harmonický signál, jehož frekvence se v čase lineárně mění. Pro případ rostoucí frekvence se může předpis vyjádřit vztahem



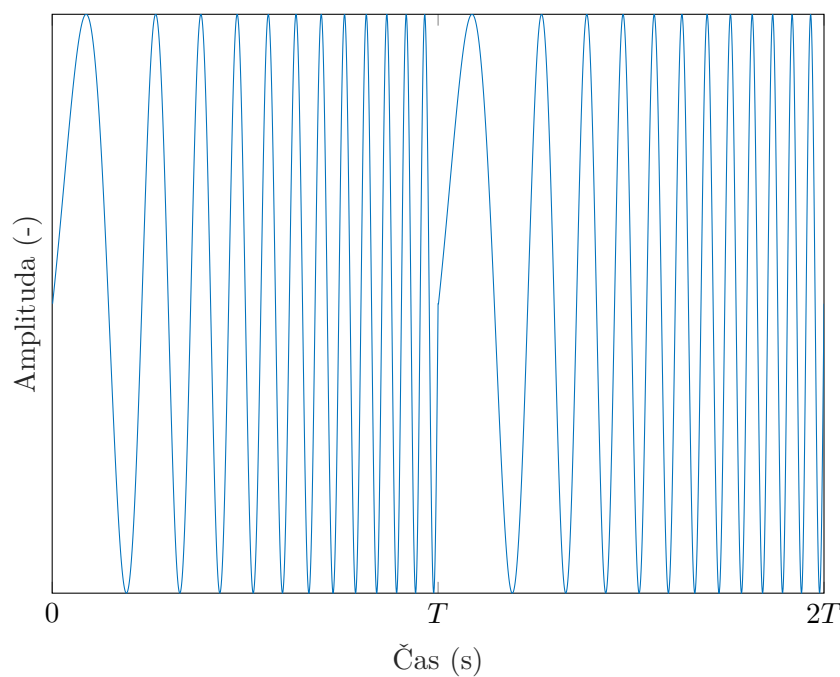
Obrázek 2.1: Blokové schéma FMCW radaru.

$$f(t) = f_c + \frac{B}{T}t, \quad (2.1)$$

kde f_c je počáteční frekvence, B šířka pásma a T délka chirpu. Modelový průběh signálu je znázorněn na obrázku 2.2.

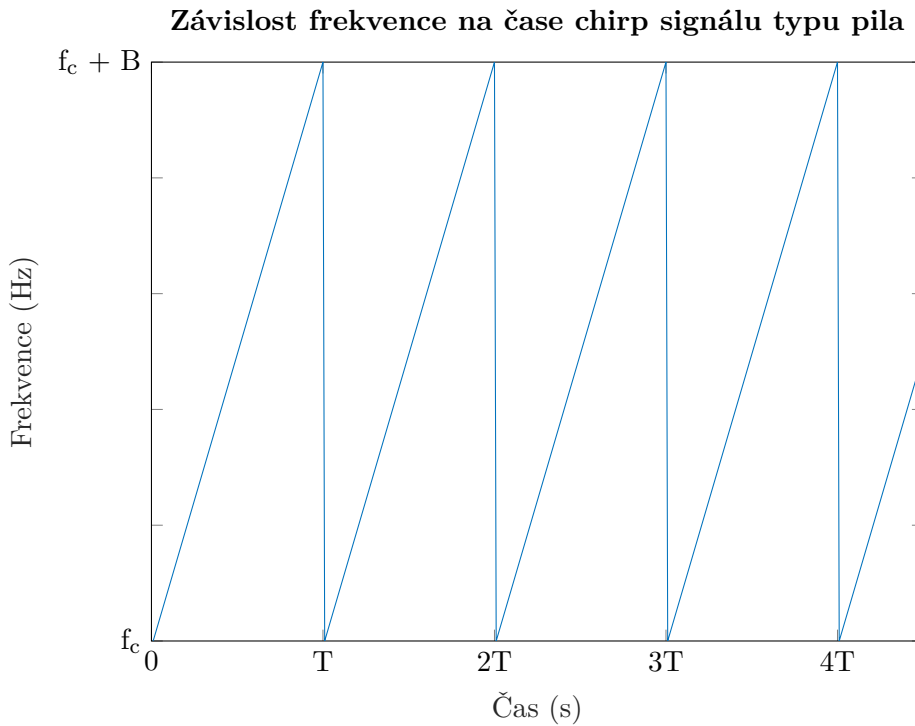
Právě poměr parametrů B a T je velmi důležitý, protože udává strmost rampy¹, na kterou je možné se dívat jako na „rychlost“ jakou frekvence vzrůstá.

Závislost amplitudy chirp signálu na čase



Obrázek 2.2: Chirp v časové oblasti pro dvě periody.

¹Rampa je lineární náběh frekvence chirpu.



Obrázek 2.3: FMCW signál pila pro čtyři period.

2.3 Princip FMCW radaru

V následující části jsou použité rovnice čerpány z literatury [4].

Vysílaný signál se skládá z posloupnosti chirpů. V této bakalářské práci je k vysvětlení problematiky analýzy signálu použit typ chirpu pila s rostoucí frekvencí, který je znázorněn na obrázku 2.3. V praxi se nepoužívá jen jeden typ modulace, protože každá má své výhody a nevýhody a jejich kombinací lze dosáhnout větší přesnosti v měření vzdálenosti i rychlosti. Zároveň je FMCW princip znázorněn na jednoduchém SISO (Single input single output) rozložení antén. To znamená, že je použita jen jedna vysílací i přijímací anténa. V sekci 2.4 je vysvětlen přístup MIMO, kde je použito více přijímacích i vysílacích antén.

Aby bylo možné vyjádřit signál, je potřeba zjistit okamžitou fázi $\mu(t)$. Ta se získá integrací změny frekvence podle času (vzorec (2.1)) jako

$$\mu(t) = 2\pi \int_0^t f(t)dt + \mu_0 = 2\pi \left(f_c t + \frac{B}{T} \frac{t^2}{2} \right) + \mu_0. \quad (2.2)$$

Nyní je možné popsat signál v prvním chirpu, začínajícím v čase $t = 0$, s předpokladem, že μ_0 je počáteční fáze chirpu

$$x_{tx} = A \cos(\mu(t)) = A \cos \left(2\pi \left(f_c t + \frac{B}{T} \frac{t^2}{2} \right) + \mu_0 \right), \quad (2.3)$$

kde $0 < t < T$.

Rovnice (2.3) nepopisuje celý vysílaný signál, ale jen první chirp. Proto je nutné provést následující substituci, aby vznikl signál, který je posloupností, za sebou jdoucích, chirpů

$$t = nT + t_a, \quad (2.4)$$

kde $0 < t_a < T$ a n je celé číslo.

Substituce (2.4) se dosadí do rovnice (2.3)

$$x_{\text{tx}} = A \cos \left(2\pi \left(f_c (nT + t_a) + \frac{B t_a^2}{T} \right) + \mu_0 \right), \quad (2.5)$$

kde $0 < n < N$ a $0 < t_a < T$.

Člen $\frac{B t_a^2}{T}$ vyjadřuje kvadratický nárůst fáze, což po derivaci znamená lineární nárůst frekvence v průběhu chirpu. Tedy, t se mění jen od 0 do T , což je délka jednoho chirpu a proto se jen t zamění za t_a .

Vzorec (2.5) je předpisem vysílaného signálu. Přijatý signál bude od vyslaného signálu zpožděn o čas τ , který je závislý na vzdálenosti cíle od radaru.

$$x_{\text{rx}} = B \cos(\mu(t - \tau)) = B \cos \left(2\pi \left(f_c (nT + t_a - \tau) + \frac{B (t_a - \tau)^2}{T} \right) \right). \quad (2.6)$$

Na první vstup směšovače je přiveden přijatý signál a na druhý vyslaný. Výstupem toho prvku je následující mezifrekvenční signál

$$x_{\text{IF}}(t) = x_{\text{tx}}(t) x_{\text{rx}}(t) = A \cos(\mu(t)) B \cos(\mu(t - \tau)). \quad (2.7)$$

Při použití známého trigonometrického vzorce

$$A \cos(\alpha) B \cos(\beta) = \frac{AB}{2} (\cos(\alpha + \beta) + \cos(\alpha - \beta)), \quad (2.8)$$

je evidentní, že $x_{\text{IF}}(t)$ se bude skládat ze dvou harmonických signálů. Ovšem první část signálu vzniklá součtem argumentů, bude mít velmi vysokou frekvenci (více než dvojnásobek nosné f_c), tudíž ho bude možné po průchodu filtrem typu dolní propust zanedbat. Výsledný $x_{\text{IF}}(t)$ bude vypadat následovně

$$x_{\text{IF}}(t) = \frac{AB}{2} \cos(\alpha - \beta) = \frac{AB}{2} \cos \left(2\pi \left(f_c (nT + t_a) + \frac{B t_a^2}{T} - f_c (nT + t_a - \tau) - \frac{B (t_a - \tau)^2}{T} \right) \right). \quad (2.9)$$

Po zjednodušení výše uvedené rovnice získáme

$$x_{\text{IF}}(t) = \frac{AB}{2} \cos \left(2\pi \left(f_c \tau + \frac{B}{T} \tau t_a - \frac{B \tau^2}{T} \right) \right). \quad (2.10)$$

Za předpokladu, že ve vzdálenosti R od radaru se nachází cíl pohybující se relativní rychlostí v (vůči radaru), od kterého se vlna odrazí a dopadne na přijímací anténu. Přijatý signál bude od vyslaného signálu zpožděn o τ , které je způsobeno dobou, za kterou vlna, pohybující se rychlostí c , urazí vzdálenost k cíli a od něj zpět k radaru.

$$\tau = \frac{2(R + vt)}{c} = \frac{2(R + v(nT + t_a))}{c}. \quad (2.11)$$

Po dosazení se rovnice (2.10) změní na

$$x_{\text{IF}}(t) = \frac{AB}{2} \cos \left(2\pi \left(f_c \frac{2(R + v(nT + t_a))}{c} + \frac{B}{T} \frac{2(R + v(nT + t_a))}{c} t_a - \frac{B}{T} \left(\frac{2(R + v(nT + t_a))}{c} \right)^2 \right) \right). \quad (2.12)$$

Rovnice se dále roznásobí na

$$x_{\text{IF}}(t) = \frac{AB}{2} \cos \left(\left(\frac{B}{T} \frac{2R}{c} + \frac{2f_c v}{c} + \frac{B}{T} \frac{2vnT}{c} - \frac{B}{T} \frac{4Rv}{c^2} - \frac{B}{T} \frac{4nTv^2}{c^2} \right) t_a + \left(\frac{2f_c v}{c} - \frac{B}{T} \frac{4Rv}{c^2} \right) nT + \frac{2f_c R}{c} + \frac{B}{T} \frac{2vt_a^2}{c} - \frac{B}{T} \frac{2R^2}{c^2} - \frac{B}{T} \frac{2v^2 n^2 T^2}{c^2} - \frac{B}{T} \frac{2v^2 t_a^2}{c^2} \right). \quad (2.13)$$

Všechny členy, ve kterých se vyskytuje c^2 budou mít velmi malý vliv a mohou být zanedbány, vzhledem k tomu, že c je rychlost světla². Hodnoty T se pohybují řádově v desítkách mikrosekund, B v jednotkách gigahertzů, n typicky kolem jednoho sta. Za předpokladu, že R je několik desítek metrů a v několik desítek metrů za sekundu, zanedbatelný vliv budou mít také členy $\frac{2f_c v}{c}$, $\frac{B}{T} \frac{2vnT}{c}$ a $\frac{B}{T} \frac{2vt_a^2}{c}$.

Výsledný vztah po zjednodušení výše uvedené rovnice bude následující

$$x_{\text{IF}}(t) = \frac{AB}{2} \cos \left(2\pi \left(\frac{B}{T} \frac{2Rt_a}{c} + \frac{2f_c vnT}{c} \right) + \frac{4\pi f_c R}{c} \right). \quad (2.14)$$

Jedná se o důležitou rovnici, ze které vychází zpracování signálu FMCW radaru. Zlomek $\frac{4\pi f_c R}{c}$ představuje jen konstantní posun fáze a v průběhu chirpu se nemění. Tudíž nepřináší žádnou užitečnou informaci.

2.3.1 Určení vzdálenosti

Zjištění, jak daleko se nachází cíl od radaru, vychází z rovnice (2.14). Jak je uvedeno v [3], je důležité najít tzv. *frekvenci zázněje* signálu. Jedná se o rozdíl

² $c = 299\,792\,458 \text{ m/s} \approx 3 \cdot 10^8 \text{ m/s}$

mezi frekvencí přijatého a vyslaného chirpu a je přímo spojena se zpožděním signálu resp. se vzdáleností cíle. Frekvence zázněje se získá derivací podle času t_a okamžité fáze signálu z rovnice (2.14) jako

$$f_b = \frac{1}{2\pi} \frac{d}{dt_a} \left[2\pi \left(\frac{B}{T} \frac{2Rt_a}{c} + \frac{2f_c v n T}{c} \right) + \frac{4\pi f_c R}{c} \right] = \frac{2BR}{Tc}, \quad (2.15)$$

z čehož je vzdálenost R vyjádřena následovně,

$$R = \frac{f_b T c}{2B}. \quad (2.16)$$

■ Rozlišení ve vzdálenosti

Jak bylo výše uvedeno, vzorec pro vzdálenost cíle je (2.16). Signál je vzorkován frekvencí f_s a doba jednoho chirpu je T . Tudíž po dobu chirpu je navzorkováno N hodnot,

$$N = f_s T. \quad (2.17)$$

Poté je na signál použita Fourierova transformace, která má stejný počet vzorků jako transformovaný signál, N . Rozsah frekvenční osy je od $-\frac{f_s}{2}$ do $\frac{f_s}{2}$ tzn. délka f_s . Vzdálenost mezi jednotlivými body na frekvenční ose Δf_b je,

$$\Delta f_b = \frac{f_s}{N}. \quad (2.18)$$

Po dosazení výrazu (2.18) do rovnice (2.16) se získá rozlišení ve vzdálenosti následovně,

$$\Delta R = \frac{\Delta f_b T c}{2B} = \frac{f_s T c}{2BN} = \frac{NTc}{2BNT} = \frac{c}{2B}. \quad (2.19)$$

■ Maximální změřitelná vzdálenost

Maximální vzdálenost R_{\max} , kterou je radar schopen změřit vychází opět z rovnice (2.16) a z toho, jakou maximální frekvenci lze zachytit. Přicházející odražený signál je zpracováván analogovým převodníkem, který signál vzorkuje s frekvencí f_s . Tudíž Shannonův-Nyquistův-Kotělnikovův teorém říká, že maximální frekvence f_b , kterou lze vzorkovat je,

$$f_{b\max} = \frac{f_s}{2}. \quad (2.20)$$

Po dosazení do rovnice (2.16) se získá R_{\max} jako,

$$R_{\max} = \frac{f_s T c}{4B}. \quad (2.21)$$

■ 2.3.2 Určení rychlosti

K měření rychlosti cíle lze opět vyjít z rovnice (2.14). Konkrétně člen $\frac{2f_c v n T}{c}$, který představuje změnu frekvence přes jednotlivé průběhy chirpu. Není možné určit rychlost cíle jen z jednoho průběhu signálu, protože výsledek by byl nejednoznačný [5]. Dopplerova frekvence f_d lze napsat

$$f_d = \frac{2f_c v}{c} \quad (2.22)$$

a z toho vyjádřena rychlost v ,

$$v = \frac{f_d c}{2f_c}. \quad (2.23)$$

■ Rozlišení v rychlosti

Rozlišení v rychlosti vychází z výrazu pro rychlost (2.23). Dopplerovská frekvence je vzorkována vždy v jednom průběhu chirpu, proto vzorkovací frekvence f_{ds} je závislá na době trvání chirpu T ,

$$f_{ds} = \frac{1}{T}. \quad (2.24)$$

Poté dochází k použití Fourierovi transformace, ovšem v druhém rozměru. Neprovádí se na chirp, ale na signál přes N chirpů (viz podsekcce 2.6.1). Frekvenční osa má tedy N bodů a vzdálenost mezi nimi Δf_{ds} je,

$$\Delta f_{ds} = \frac{f_{ds}}{N} = \frac{1}{NT}. \quad (2.25)$$

Po dosazení Δf_{ds} do rovnice (2.23) se získá rozlišení v rychlosti,

$$\Delta v = \frac{\Delta f_d c}{2f_c} = \frac{c}{2f_c N T}. \quad (2.26)$$

■ Maximální změřitelná rychlost

V tomto odvození se opět vyjde z rovnice (2.23). Tentokrát maximální frekvence, kterou lze určit závisí na délce chirpu T . Jak bylo uvedeno výše v sekci 2.3.2,

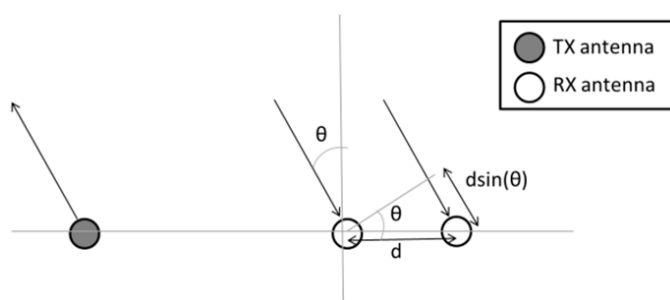
$$f_{ds} = \frac{1}{T}. \quad (2.27)$$

Opět Shannonův-Nyquistův-Kotělnikovův teorém říká, že maximální frekvence f_d , kterou lze vzorkovat je,

$$f_d = \pm \frac{f_{ds}}{2} = \pm \frac{1}{2T}. \quad (2.28)$$

Po dosazení do rovnice (2.23) se získá maximální změřitelná rychlost v_{\max} ,

$$v_{\max} = \pm \frac{f_{ds} c}{4f_c} = \pm \frac{c}{4T f_c}. \quad (2.29)$$



Obrázek 2.4: Určení azimutu s použitím dvou antén [6].

2.3.3 Určení azimutu

Azimut je další velmi důležitá informace, na kterou se klade velký důraz z hlediska přesnosti. Při použití radaru v automobilech je kritické rozlišit, jestli se cíl, který byl detekován, nachází v dráze vozidla, a zdali je tedy nutné reagovat (např. vozidlo zastavit, nebo vykonat úhybný manévr).

Určení azimutu se zakládá na principu, že se elektromagnetické vlnění, o vlnové délce λ , šíří v prostoru jako rovinná vlna (obrázek 2.4). To znamená, že při dopadu na dvě antény, které jsou od sebe vzdáleny d , bude muset signál na druhé anténě urazit navíc vzdálenost $d \sin(\theta)$, což po přepočítání na fázi vychází následovně,

$$\omega = \frac{2\pi d}{\lambda} \sin(\theta). \quad (2.30)$$

Vyjádřeno pro azimut,

$$\theta = \arcsin\left(\frac{\omega \lambda}{2\pi d}\right). \quad (2.31)$$

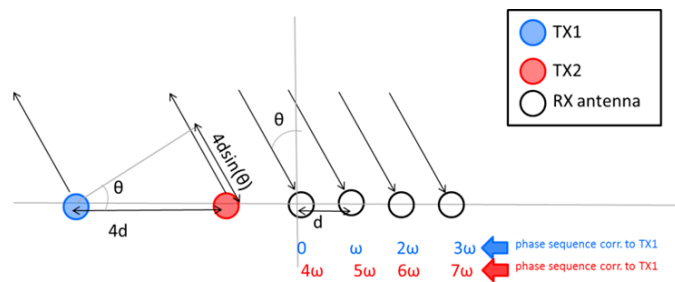
Při separaci přijímacích antén o $d = \frac{\lambda}{2}$, což je běžná hodnota, tak po dosažení do rovnice (2.31) lze získat rozsah dosažitelného azimutu,

$$\theta = \arcsin\left(\frac{\omega \lambda}{2\pi \frac{\lambda}{2}}\right) = \arcsin\left(\frac{\omega}{\pi}\right). \quad (2.32)$$

Z rovnice (2.32) plyne, že ω může nabývat hodnot $-\pi \leq \omega \leq \pi$. To odpovídá azimutu $-90^\circ \leq \theta \leq 90^\circ$.

2.3.4 Určení elevace

Nalezení elevace cíle používá stejný základ jako hledání azimutu. V obou případech jde o nalezení úhlu, ve kterém přichází vlna k radaru. Jediný rozdíl je, že azimut je určen v horizontálním pohledu a elevace je ve vertikálním. Tzn. je potřeba jiné uspořádání přijímacích a vysílacích antén.

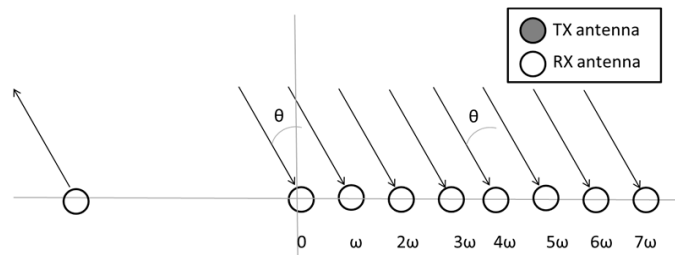


Obrázek 2.5: Princip MIMO [6].

2.4 Princip MIMO

Podle [6], MIMO je obecně zařízení s více vstupy a více výstupy. V případě radaru se jedná o zařízení s více vysílacími a přijímacími anténami. Rozlišení v úhlu radaru s N_{TX} vysílacích antén a N_{RX} přijímacích antén je ekvivalentní s radarem SIMO (Single input single output)³ (single-input-multiple-output), který má $N_{TX} \times N_{RX}$ přijímacích antén. Uspořádání MIMO přináší efektivní způsob vylepšení úhlového rozlišení.

Pokud jsou např. čtyři přijímací antény uspořádané vedle sebe ve vzdálenosti d a dvě vysílací antény od sebe vzdáleny $4d$ vznikne tzv. *virtuální anténní pole* (obrázek 2.5). Tato soustava se bude chovat jako pole s jednou vysílací a osmi přijímacími anténami poskládanými vedle sebe jako na obrázku 2.6. Na obrázku 2.5 musí již při vysílání signál z červené antény urazit navíc dráhu $4d \sin(\theta)$, kde θ je azimut cíle, oproti vyslanému signálu z modré antény, tudíž přijatý odražený signál na nejbližší anténě bude zpožděn o 4ω , což je stejné jako zpoždění páté antény na obrázku 2.6.



Obrázek 2.6: Demonstrace anténního pole bez principu MIMO [6].

Také je možné tímto způsobem vytvořit dvoudimenzionální virtuální anténní pole, které umožní současně s azimutem měřit i elevaci. K tomu je potřeba alespoň jednu vysílací anténu posunout vertikálně (mimo řadu ostatních antén) jako na obrázku 2.7.

³Jedná se o uspořádání, kde je použita jen jedna vysílací anténa, ale více přijímacích antén.



Obrázek 2.7: Dvoudimenzionální virtuální anténní pole [6].

2.4.1 Multiplex

Použití MIMO přináší problém rozlišení signálů z jednotlivých vysílacích antén. Aby bylo možné využít výhody virtuálního anténního pole, je nutné na přijímacích anténách odlišit, z jakých vysílacích antén signál přichází. K tomu se používají různé typy ortogonálních signálů.

Nejčastěji používaným je časový multiplex, který vychází z toho, že vždy vyzařuje signál jen jedna vysílací anténa. Další anténa začne vyzařovat až poté, co předchozí vysílání ukončila. Tím je dosaženo, že signály jsou odlišitelné a mezi sebou neinterferují. Jedná se o metodu s jednoduchou implementací ovšem není zcela efektivní, jelikož vždy vysílá jen jedna anténa

Dalším možným multiplexem je fázový posun vysílaného signálu. Jedná se o jednoduchý princip, kdy vysílací antény vysílají signál s různým posunem a tím lze na straně přijímačů rozlišit, jaký signál přišel z jaké antény. Jde o efektivnější metodu než časový multiplex, jelikož mohou vysílat všechny antény najednou.

2.5 Efektivní odrazná plocha

V literatuře o radarové technice se často používá zkratka RCS (Radar cross section), neboli efektivní odrazná plocha. Jedná se o pojem, který popisuje schopnost objektu odrážet elektromagnetické záření, což je ovlivněno mnoha faktory (např. tvarem, rozměry, použitým materiálem, frekvencí vlnění a další).

Při zavádění tohoto termínu se předpokládá, že radar je od cíle dostatečně daleko, aby se dopadající vlna dala považovat za sférickou, jejíž amplituda klesá s druhou mocninou vzdálenosti. A také se předpokládá, že objekt, na který vlna dopadá, je dostatečně malý a vzdálený od radaru, aby se odražená vlna dala považovat za sférickou [7].

Efektivní odrazná plocha vychází z porovnávání dvou plošných hustot energie. Jedná se o hustotu energie dopadající vlny změřené u cíle a plošnou hustotu energie odražené vlny od cíle zachycené radarem ve vzdálenosti R . Plošná hustota energie elektromagnetické vlny je definována jako,

$$W = \frac{EH}{2} = \frac{Y_0|E|^2}{2}, \quad (2.33)$$

kde E je intenzita elektrického pole, H intenzita magnetického pole a Y_0 je admitance volného prostoru. Jednotky těchto veličin jsou V/m, A/m a S. Pokud elektromagnetická vlna dopadá na objekt o ploše σ , výkon P_1 , který objekt odrazí se získá vynásobením rovnice (2.33) plochou σ ,

$$P_1 = \sigma W_1 = \sigma \frac{Y_0 |E_1|^2}{2}, \quad (2.34)$$

kde W_1 je plošná hustota energie v místě cíle a E_1 je elektrická intenzita přicházející vlny.

Další předpoklad je, že odražená vlna je emitována objektem rovnoměrně do všech směrů. Poté lze vyjádřit hustotu odražené vlny W_2 jako výkon odražené vlny P_1 snížen úměrně vzdáleností R , ze které je vlna pozorována,

$$W_2 = \frac{P_1}{4\pi R^2} = \frac{\sigma Y_0 |E_1|^2}{8\pi R^2}. \quad (2.35)$$

Obdobně jako v rovnici (2.33) se dá hustota energie odražené vlny také vyjádřit jako,

$$W_2 = \frac{E_2 H_2}{2} = \frac{Y_0 |E_2|^2}{2}, \quad (2.36)$$

kde E_2 a H_2 jsou veličiny v místě přijímače. Výsledný vztah pro efektivní odraznou plochu se získá položením do rovnosti výrazů (2.35) a (2.36) a vyjádřením σ jako,

$$\sigma = 4\pi R^2 \frac{|E_2|^2}{|E_1|^2}. \quad (2.37)$$

Pro lepší vizualizaci se hodí představa, že objekt s určitou efektivní odraznou plochou, má stejné schopnosti odrážet elektromagnetickou vlnu, jako sféra z dokonalého vodiče o stejném průřezu, jako je efektivní odrazná plocha objektu [7]. V tabulce 2.1 jsou vypsané orientační hodnoty efektivní odrazné plochy pro různé objekty. Podle rovnice (2.37) efektivní odrazná plocha není závislá na frekvenci, ale v reálném prostředí tomu tak neplatí kvůli útlumu volného prostoru, který roste s kvadrátem frekvence. Závislost je zřetelná z tabulky 2.2. Efektivní odrazná plocha se také mění v závislosti na polarizaci elektromagnetické vlny, což vytváří velké rozpětí hodnot. Další významný vliv může mít to, jak je měřený předmět v prostoru natočen.

■ 2.5.1 Koutové odražeče

Koutový odražeč je pasivní zařízení, které má díky své geometrické stavbě velmi vysokou efektivní odraznou plochu. Jedná se o tři navzájem kolmé plochy, které odrazí přicházející paprsek respektive vlnu vždy zpět do směru odkud přišla (obrázek 2.8).

Tato zařízení mají velmi široké využití. Např. jsou umístěny na Měsíci k měření vzdálenosti mezi Zemí a Měsícem pomocí laseru. U radarů se používají k experimentálním měřením a hlavně při kalibraci. Koutovým odražečem se pohybuje před radarem a sbírají se data, aby se poté reálné

Objekt	RCS (m^2)
„Neviditelný“ letoun F-117A	0,025
Pták	0,1
Lidské tělo	1
Malé jednomotorové letadlo	2
Větší stíhačka	6 – 10
Velké komerční letadlo	40 – 100
Koutový odražeč (široký 30 cm)	500

Tabulka 2.1: Tabulka RCS v mikrovlnném pásmu pro různé objekty [10].

Frekvence (MHz)	σ (m^2)
410	0,033 – 2,33
1120	0,098 – 0,997
2890	0,140 – 1,05
4800	0,368 – 1,88
9375	0,495 – 1,22

Tabulka 2.2: Hodnoty efektivní odrazné plochy člověka na různých kmitočtech mikrovlnného pásma [10].

hodnoty z provozu porovnávaly s těmi kalibračními a tím se zjistila poloha cíle.

2.6 Metody zpracování dat

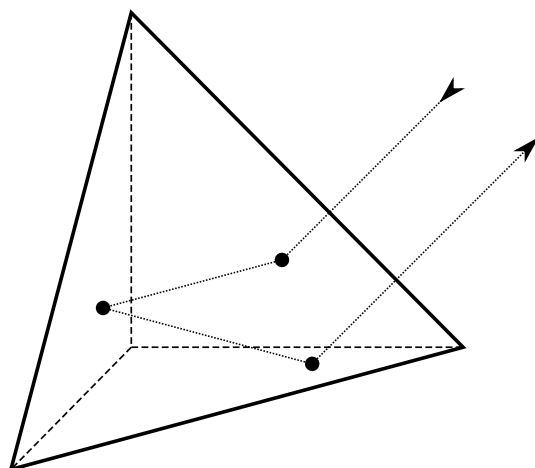
2.6.1 2D FFT

Jde o metodu, která využívá DFT (Discrete Fourier transform) k analýze signálu, konkrétněji k nalezení frekvence, na které přijatý signál kmitá. Pro snížení výpočetní náročnosti se používá FFT (Fast Fourier transform), která dává stejné výsledky s menším počtem potřebných operací.

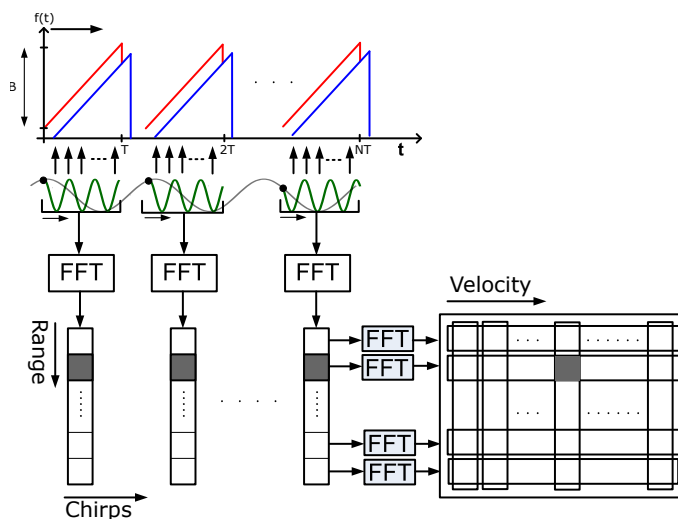
Navzorkované signály z přijímací antény se poskládají do matice, kde do každého sloupce se uloží jeden úsek mezifrekvenčního signálu z přijatého chirpu. Počet řádků tedy představuje počet vzorků v jednom chirpu. Poté se na matici použije první FFT ve vertikálním směru (obrázek 2.9). Ve sloupcích se tedy zobrazí místo signálu v čase spektrální osa chirpu. Na ní lze pozorovat jakou frekvenci má přijatý signál. Jedná se o frekvenci zázněje, z které je možné jednoduchým přepočtem (sekce 2.3.1) získat vzdálenost.

Nyní následuje druhé FFT, které je provedeno v horizontálním směru na vzorky spektra přijatého signálu. Touto analýzou se získá Dopplerovská frekvence, ze které lze určit rychlost pohybu cíle (sekce 2.3.2).

Výsledkem této metody je tzv. Range-Doppler plot, neboli graf, který zobrazuje v jaké vzdálenosti se nachází cíl a jakou rychlostí se pohybuje. Může vypadat například jako na obrázku 2.10.



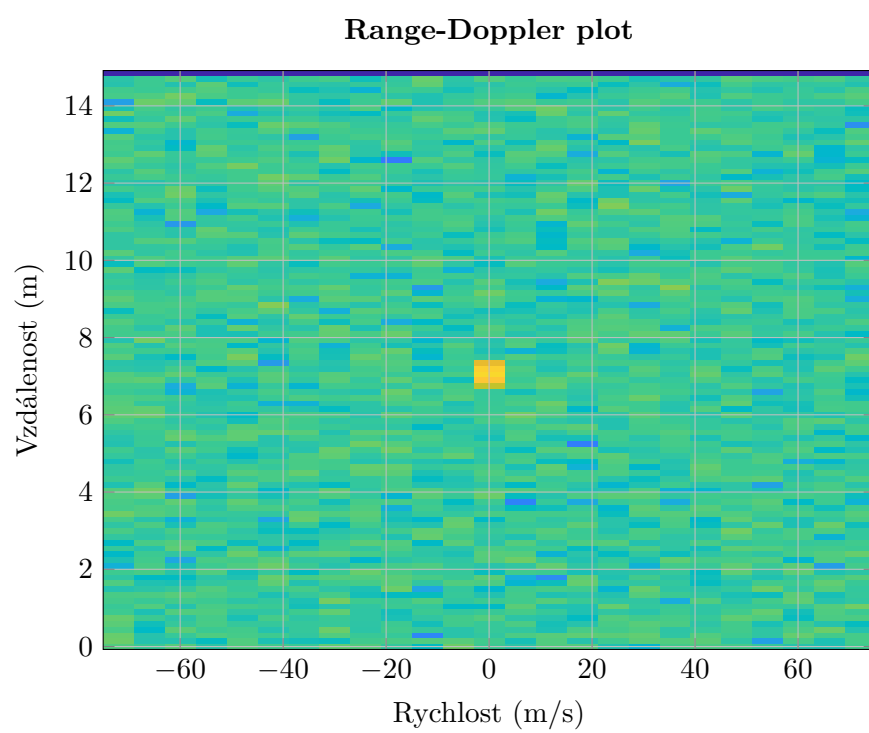
Obrázek 2.8: Odraz paprsku v koutovém odražeči.



Obrázek 2.9: 2D FFT [5].

2.6.2 Super resolution metody

Jedná se o algoritmy, které se výhradně používají k určování azimutu a elevace. Jedná se například o metodu MUSIC (Multiple signal classification) nebo ESPRIT (Estimation of signal parameters by rotational invariance techniques). Tyto algoritmy jsou schopné dosáhnout velmi vysokého rozlišení i z malého počtu vzorků, nicméně jejich výpočetní náročnost je velmi vysoká, tudíž se často využívají různě modifikované verze [8].



Obrázek 2.10: Range-Doppler plot.

Kapitola 3

Praktická část

Tato část bude zaměřena na využití znalostí popsaných v kapitole 2. Z firmy Valeo autoklimatizace, k.s. byla poskytnuta data z automobilu VW Crafter, na kterém byly nainstalovány čtyři radary. Auto se rozjelo a pokračovalo několik metrů konstantní rychlostí kolem stacionárních objektů.

V prostředí Mathworks MATLAB naprogramuji zpracování těchto dat a jejich vyhodnocení. Následně mé výsledky porovnáám s těmi, které byly vyhodnoceny přímo v mikroprocesorech radarů.

3.1 Charakteristika dat a testovací jízdy

Data jsou dostupná v CSV (Comma-separated values) souboru, jehož struktura je vysvětlena v sekci 3.1.3. Nejedná se o veškeré vzorky mezifrekvenčních signálů, ale již o potenciální cíle. Tento výběr se provádí pomocí CFAR (Constant false alarm rate) algoritmu.

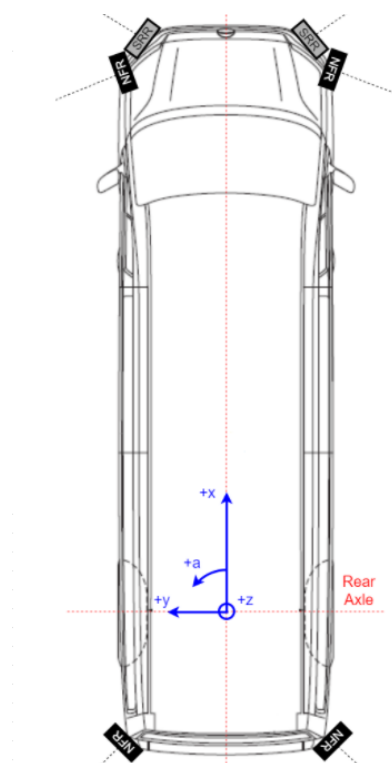
Tento algoritmus se velmi často využívá v radarových systémech. Slouží k odlišování reálných cílů od těch falešných, vyvolaných vysokou úrovní šumu. Algoritmus adaptivně nastavuje testovací úroveň, podle které rozhoduje. Tato úroveň se určuje zejména podle odstupu signálu od šumu. V průběhu je upravována, aby se dosahovalo konstantní chybovosti. To znamená, že pokud bude rozhodovací úroveň nastavena příliš nízko, tak mezi cíle vyhodnocené algoritmem jako reálné se dostane velká část falešných, ale i velká část reálných cílů bude detekována. Pokud se úroveň nastaví vyšší, tak se sníží počet falešných cílů, ale zvýší se počet reálných cílů, které nebyly nalezeny [9].

Ke každému radaru je také dostupná datová struktura obsahující kalibrační data (sekce 3.1.4) a FoV (Field of view), neboli zorné pole (sekce 3.2.2).

3.1.1 Rozmístění radarů

Radary jsou na vozidle umístěny čtyři. Nachází se v předním a zadním nárazníku, na krajích (na obrázku 3.1 s označením NFR).

Délka vozidla je 6836 mm a jeho šířka 1832 mm. Střed souřadnic se nachází uprostřed zadní nápravy. Nulový azimut se vede středem vozidla směrem



Obrázek 3.1: Rozložení radarů na vozidle (Zdroj Valeo autoklimatizace, k.s.).

	Vpředu vlevo	Vpředu vpravo	Vzadu vpravo	Vzadu vlevo
RadarPosition	0	1	2	3
Azimut (°)	110	-110	-135	135
x (mm)	4977,5	5000,2	-1154,0	-1152,5
y (mm)	1029,7	-1029,9	-976,1	976,6
z (mm)	865,0	870,0	495,0	495,0

Tabulka 3.1: Rozmístění radarů na vozidle v souřadnicích vůči středu zadní osy, posun v azimutu a označení jednotlivých radarů.

dopředu a kladný úhel se uvádí proti směru hodinových ručiček. Pozice jednotlivých radarů a jejich natočení jsou uvedeny v tabulce 3.1.

■ 3.1.2 Anténní pole

U radaru je využito principu MIMO s časovým multiplexem. Jsou použity dvě vysílací a tři přijímací antény (označení TX0 a TX2 pro vysílací a RX0, RX1 a RX2 pro přijímací). Tudíž virtuální anténní pole se skládá z šesti kombinací antén.

■ 3.1.3 Struktura dat

Data v CSV souboru jsou zpracovány do tabulky, jako na obrázku 3.2. V každém řádku je jedna detekce cíle a k ní potřebné informace. Položky v jednot-

	1	2	3	4	5	6	7
	CycleNumber	RadarPosition	CoGDoppler_mps	CoGRange_m	SNR_half_dB	Azimuth_rad	Elevation_rad
1	64303	0	0.0117	1.2148	42	2.3178	1.8413
2	64303	0	0.0078	1.3164	44	2.3195	1.8186
3	64303	0	0.0039	1.4375	46	3.1102	1.1484
4	64303	0	0.0078	9.9219	74	1.5219	1.5167
5	64303	0	0.0078	9.9961	71	1.4992	1.4626
6	64303	0	0.0078	10.1367	63	1.4713	1.4207
7	64303	0	0.0039	10.2539	69	1.4294	1.5202
8	64303	0	0.0039	10.3516	55	1.5027	1.5551
9	64303	0	0.0078	10.5000	46	1.5167	1.5359
10	64303	0	0.0039	10.6406	68	1.0804	1.5481

	7	8	9	10	11	12	13	14
	Elevation_rad	Lateral_m	Longitudinal_m	Heigth_m	TX0_RX0_real	TX0_RX0_imag	TX2_RX0_real	TX2_RX0_imag
1	1.8413	4.1816	1.8890	1.1847	7	11	5	-13
2	1.8186	4.1083	1.9649	1.1829	-31	-15	11	2
3	1.1484	3.6665	1.0712	0.2707	-4	10	8	3
4	1.5167	5.4610	10.9255	0.3234	716	696	-441	57
5	1.4626	5.6875	10.9422	-0.2196	-190	-767	488	145
6	1.4207	5.9725	11.0032	-0.6558	115	657	-152	-29
7	1.5202	6.4199	11.1686	0.3412	379	-368	-75	381
8	1.5551	5.6810	11.3563	0.6974	-77	128	-8	-161
9	1.5359	5.5445	11.5082	0.4936	163	66	-85	-13
10	1.5481	9.9876	10.4140	0.6186	293	675	121	-262

Obrázek 3.2: Ukázka dodaných dat před zpracováním.

livých sloupcích jsou postupně popsány.

■ RadarPosition

Číslo od nuly do tří. Jde o identifikátor radaru (tabulka 3.1).

■ CoGDoppler_mps

Rychlost cíle vypočtená z Dopplerovské frekvence metodami a algoritmy společnosti Valeo autoklimatizace, k.s. udávaná m/s. Počítá se jako relativní rychlost vůči radaru.

■ CoGRange_m

Vzdálenost cíle vypočtená metodami a algoritmy společnosti Valeo autoklimatizace, k.s. udávaná v metrech. Vzdálenost je měřená vzhledem k danému radaru. Některé detekce mohou být jen odrazy od nárazníku, a jejich vzdálenost je v řádu několika centimetrů. Tyto cíle je tedy možno ignorovat.

■ SNR_half_dB

Odstup signálu od šumu v jednotkách dB. Dle toho parametru je možné detekce filtrovat. Cíle s příliš vysokým SNR (Signal to noise ratio) budou nejspíše opět odrazy od nárazníku a příliš nízké SNR znamená, že signál je znehodnocen šumem, tudíž bude nepřesný.

■ Azimuth_rad

Získaný azimut společností Valeo autoklimatizace, k.s. v radiánech, vyjádřený vůči vozidlu. S touto hodnotou budou porovnávány výsledky získané z této práce.

■ Elevation_rad

Elevace určená metodami společnosti Valeo autoklimatizace, k.s. v radiánech vůči radaru. Hodnota 90° nebo $\pi/2$ odpovídá cíli přímo před radarem v horizontální rovině. S touto hodnotou se budou opět porovnávat výsledky této práce.

■ Lateral_m

Souřadnice x v metrech a v souřadnicovém systému vozidla. Hodnota je přepočtená ze vzdálenosti a azimutu vůči radaru.

■ Longitudinal_m

Souřadnice y v metrech a v souřadnicovém systému vozidla. Hodnota je opět přepočtená ze vzdálenosti a azimutu vůči radaru.

■ Height_m

Výška v metrech, ve které se nachází detekovaný cíl vůči středu zadní osy vozidla.

■ Komplexní data

Samotná nezpracovaná data z jednotlivých antén. Hodnoty jsou vždy pojmenovány podle toho, pro jakou kombinaci vysílací a přijímací antény byly změřeny. Např. TX2_RX0 je kombinace pro vysílací anténu TX2 a přijímací anténu RX0. Hodnoty jsou zvláště rozděleny na reálnou (TX2_RX0_real) a imaginární část (TX2_RX0_imag). Vysílací anténa TX2 vysílá střídavě signál s fází 0° (TX2_0_RX0) a 180° (TX2_180_RX0). To ovšem nepřináší žádnou novou informaci o cíli, takže je možné tyto hodnoty sečíst a zprůměrovat.

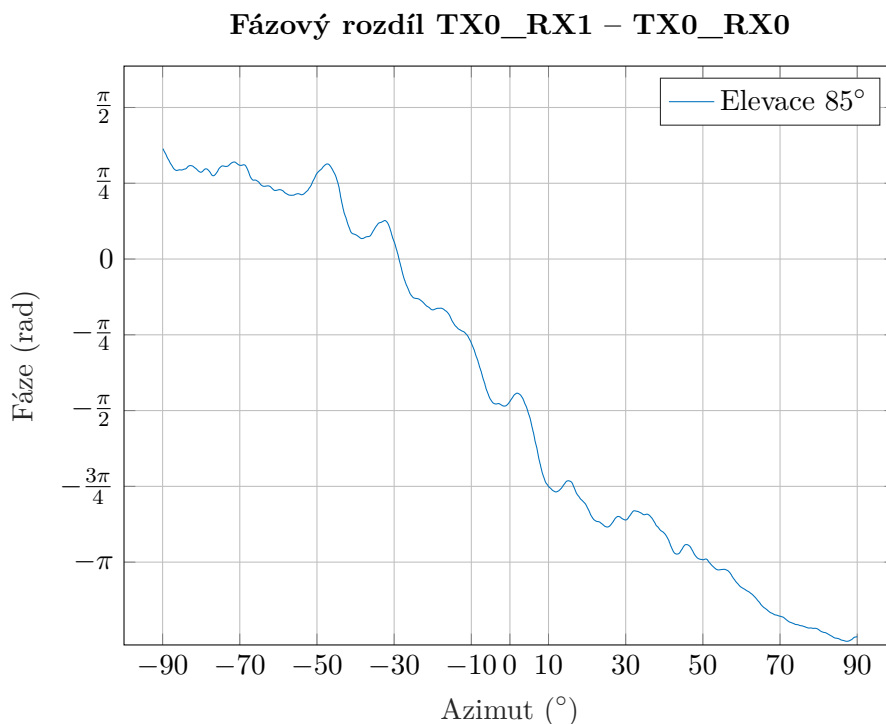
Z těchto dat se pomocí metody využívající kalibrační křivku (sekce 3.1.4) získá informace o azimutu a elevaci.

■ 3.1.4 Kalibrační křivky

U tohoto radaru je k určení azimutu a elevace použita kalibrační křivka. Azimut a elevace se určuje podle fázového rozdílu mezi jednotlivými přijímacími anténami (2.3.3). Kalibrační křivka (obrázek 3.3) je soubor experimentálně naměřených hodnot fázových rozdílů pro každou kombinaci antén. Tyto křivky se obecně měří pomocí umělého cíle, často se využívá koutový odražeč (sekce 2.5.1), kterým se před radarem pohybuje pro různé elevace a azimuty

a zaznamenávají se fázové rozdíly mezi referenční dvojicí antén (v tomto konkrétním případě TX0_RX0) a ostatními.

Matice kalibračních křivek má rozměr $5 \times 11 \times 1801$, pro 5 kombinací přijímací a vysílací antény, 11 kroků v elevaci (65° do 115° s krokem 5°) a 1801 kroků v azimutu (-90° až 90° s krokem $0,1^\circ$).



Obrázek 3.3: Kalibrační křivka pro kombinaci antén TX0_RX1.

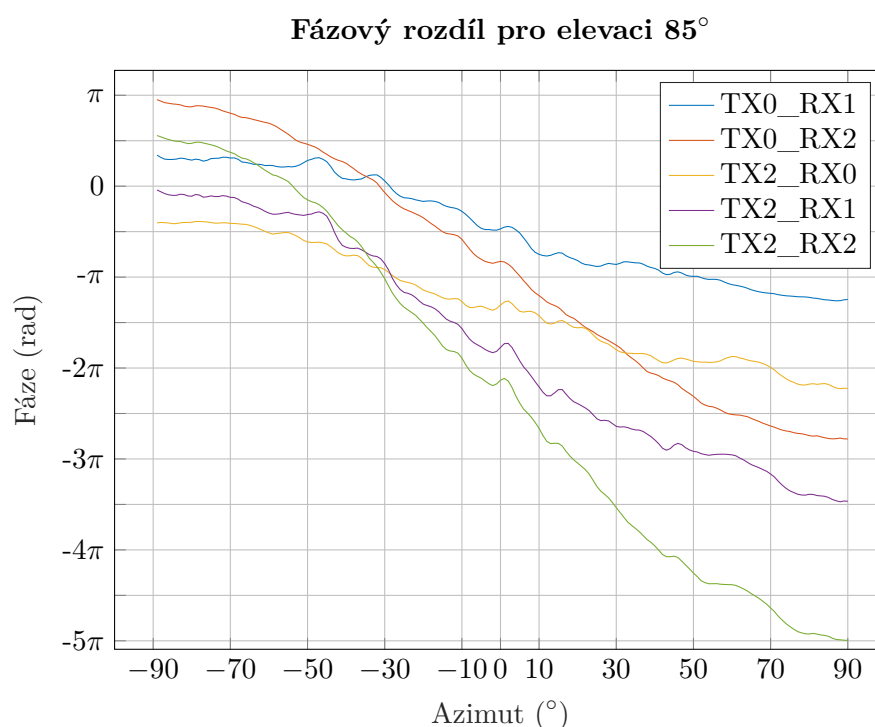
Na obrázku 3.3 je vidět, že pro jeden fázový rozdíl je možné získat více azimutů. Tato nejednoznačnost se dá lehce odstranit použitím více kombinací antén.

Obrázek 3.4 zobrazuje průběh fázových rozdílů pro všech pět kombinací antén pro elevaci 85° . Je zde vidět, že při použití více antén již nedochází k nejednoznačnosti.

3.2 Zpracování dat

Tato část se zabývá zpracováním dat. To bylo implementováno v prostředí MathWorks MATLAB. Cílem této sekce je vysvětlit a ukázat princip, jakým se z komplexních vzorků získá informace o azimutu a elevaci. Získané výsledky bude následně možné porovnat s daty, která byla určena pomocí metod společnosti Valeo autoklimatizace, k.s.

Algoritmus se skládá z několika částí. Zaprvé je potřeba z komplexních čísel získat informaci o fázi pro každou kombinaci vysílací a přijímací antény. Následně je nutné porovnat fázové rozdíly mezi referenční anténou a ostatními



Obrázek 3.4: Kalibrační křivka pro všechny kombinace antén.

anténami a kalibračními křivkami. Kalibrační křivky obsahují pro jednotlivé úrovně elevace a azimutu patřičné fázové rozdíly. Nakonec azimut a elevace vůči anténě se určí nalezením nejbližší shody mezi fázovými rozdíly detekce a kalibrační křivky.

Výstupem zpracování bude rekonstrukce jízdy automobilu pomocí 3D vykreslení cílů v prostoru okolo vozidla.

3.2.1 Příprava a filtrace

Prvním krokem je načtení dat z CSV souboru a uložení do proměnné `raw_data`. Dále je provedena filtrace podle SNR, protože detekce se SNR vyšším než 90 dB jsou většinou jen odrazy od nárazníků, pod kterým jsou radary umístěny. Detekce s nižším SNR než 50 dB mohou být nechtěné odrazy od vozovky, či jiné nepotřebné odrazy. Posledním kritériem je počáteční cyklus detekcí. Jak je vidět na obrázku 3.5 proměnná `sorted_data` obsahuje vyfiltrované detekce, které se následně používají k dalšímu zpracování. Do proměnné `new_data`, která je také typu `table`, jsou ukládány všechny vypočítané informace o detekci.


```

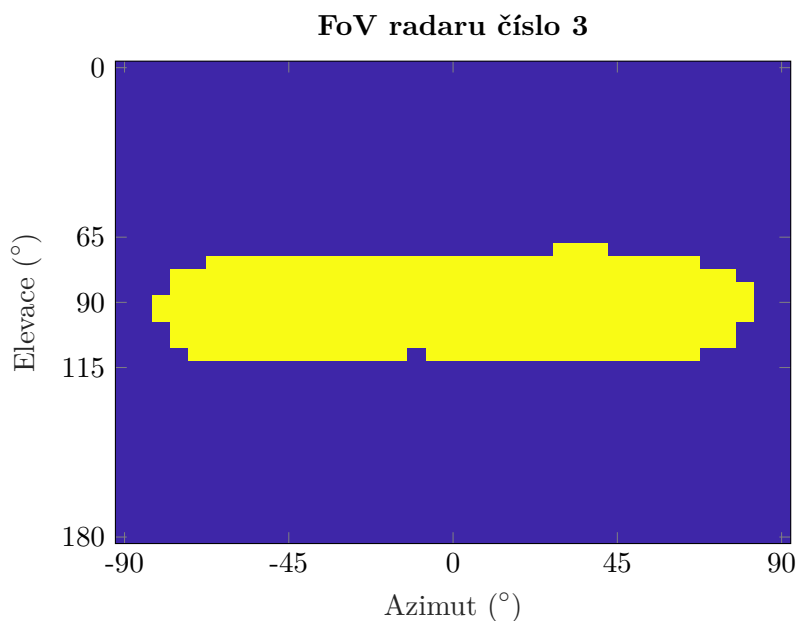
1   low_SNR = 50;           % spodni mez filtrace podle SNR
2   high_SNR = 90;         % horni mez filtrace podle SNR
3   first_cycle = 64302;   % cyklus od ktereho se zacinaji ...
                             zpracovavat data
4   sorted_data = raw_data(raw_data.SNR_half_dB > low_SNR ...
5   & raw_data.SNR_half_dB < high_SNR, :);
6   sorted_data = sorted_data(sorted_data.CycleNumber ≥ ...
                             first_cycle, :);

```

Obrázek 3.5: Ukázka filtrace dat.

3.2.2 FoV radarů

Každý radar má omezené FoV, kde je schopný určovat cíle. Teoreticky je možné rozlišovat objekty v azimutu od -90° do 90° . V praxi ale není možné takového rozsahu dosáhnout a se ve skutečnosti pohybuje zhruba od -70° do 70° . Elevace je samozřejmě také omezená, nicméně tu není potřeba ve velkém rozpětí měřit a většinou vystačí $\pm 20^\circ$. Samozřejmě záleží na účelu, ke kterému je radar určen. Pokud se jedná o radar na dlouhé vzdálenosti, v elevaci stačí menší rozsah. Naopak, pokud se jedná o radar na krátké vzdálenosti používaný například při parkování, je důležité rozlišovat i objekty blízko vozidla.



Obrázek 3.6: Matice FoV radaru 3. Žlutá barva reprezentuje zorné pole, kde jsou zjištěné cíle validní.

Každý z použitých radarů má svou vlastní matici představující FoV, která má rozměry 37×37 (od 0° do 180° v elevaci a od -90° do 90° v azimutu). Obrázek 3.6 právě reprezentuje matici FoV pro radar číslo 3. Žlutá část představuje zorné pole radaru, kde jsou detekce validní. Mimo tuto

část, tj. modrá barva, mohou vznikat falešné cíle z důvodu nejednoznačnosti kalibračních křivek. Proto je nutné při vyhledávání azimutu a elevace určit, jestli se cíl nachází ve FoV, či nikoli.

Na obrázku 3.7 je ukázána část kódu, kde je provedena interpolace původní matice FoV2d_37, která má právě rozměr 37×37 . Z ní jsou vybrány pouze řádky 14 až 24, což odpovídá rozměru matice kalibračních dat ($65^\circ - 115^\circ$). Matice FoV2d má rozměry $4 \times 11 \times 1801$. 11 různých elevací a 1801 různých azimutů pro čtyři radary. Matice FoV2d obsahuje nulové hodnoty na pozicích, které jsou mimo FoV a jedničky na pozicích ve FoV. Jedná se tedy o masku, kterou se omezí matice, v níž se hledá azimut a elevace (obrázek 3.12) jen na pozice, kde je možné cíle detekovat.

```

1     [X, Y] = meshgrid(linspace(1,37,1801), linspace(1, 11, 11));
2
3     FoV2d(1, :, :) = interp2(squeeze(FoV2d_37(1,14:24, :)), X, ...
4         Y, 'nearest');
5     FoV2d(2, :, :) = interp2(squeeze(FoV2d_37(2,14:24, :)), X, ...
6         Y, 'nearest');
7     FoV2d(3, :, :) = interp2(squeeze(FoV2d_37(3,14:24, :)), X, ...
8         Y, 'nearest');
9     FoV2d(4, :, :) = interp2(squeeze(FoV2d_37(4,14:24, :)), X, ...
10        Y, 'nearest');
```

Obrázek 3.7: Ukázka kódu interpolace FoV matic pro jednotlivé radary.

■ 3.2.3 Výpočetní cyklus

Samotný postup k získání azimutu a elevace pro dostupné detekce se skládá z cyklu `for r = 1:rows`, kde `r` je proměnná nabývající hodnot od jedné do `rows`, což je počet zpracovávaných detekcí.

■ Fáze signálů

Prvním krokem v cyklu je převod vzorků z tabulky `sorted_data`, kde jsou odděleně uloženy reálné a imaginární části, na komplexní čísla a následný převod do exponenciálního tvaru, ze kterého je užitečná pouze informace o fázi komplexního čísla (ukázka této operace je na obrázku 3.8). Fáze z jednotlivých dvojic antén je uložena do připravených proměnných v tabulce `new_data`.

```

1   new_data.TX0_RX0(r) = angle(sorted_data.TX0_RX0_real(r) ...
    + 1i*sorted_data.TX0_RX0_imag(r));
2   new_data.TX2_RX0(r) = angle(sorted_data.TX2_RX0_real(r) ...
    + 1i*sorted_data.TX2_RX0_imag(r));
3   new_data.TX0_RX1(r) = angle(sorted_data.TX0_RX1_real(r) ...
    + 1i*sorted_data.TX0_RX1_imag(r));
4   new_data.TX2_RX1(r) = angle(sorted_data.TX2_RX1_real(r) ...
    + 1i*sorted_data.TX2_RX1_imag(r));
5   new_data.TX0_RX2(r) = angle(sorted_data.TX0_RX2_real(r) ...
    + 1i*sorted_data.TX0_RX2_imag(r));
6   new_data.TX2_RX2(r) = angle(sorted_data.TX2_RX2_real(r) ...
    + 1i*sorted_data.TX2_RX2_imag(r));

```

Obrázek 3.8: Převod komplexních vzorků signálu na algebraický tvar a následné určení fáze.

■ Násobení kalibračních křivek

Dalším krokem je v kalibračních datech nalézt co nejpodobnější kombinaci fázových rozdílů všech dvojic antén. To znamená pro každou dvojici přijímací a vysílací antény najít, (ideálně stejný) prakticky, co nejpodobnější fázový rozdíl od referenční antény a z něj určit azimut a elevaci. Následně se porovnají azimuty a elevace od jednotlivých antén a nějakým způsobem se vybere (např. průměrováním) konečný azimut a elevace vůči radaru.

```

1   multi_matrix = ...
2   exp(1i*new_data.TX0_RX0(r))*conj(exp(1i*zeros(11,1801))) ...
3   + ...
4   exp(1i*new_data.TX2_RX0(r))* ...
5   conj(exp(1i*squeeze(calib_curve(RadarPosition(r)+1,4,(:, :)))) ...
    + ...
6   exp(1i*new_data.TX0_RX1(r))* ...
7   conj(exp(1i*squeeze(calib_curve(RadarPosition(r)+1,1,(:, :)))) ...
    + ...
8   exp(1i*new_data.TX2_RX1(r))* ...
9   conj(exp(1i*squeeze(calib_curve(RadarPosition(r)+1,5,(:, :)))) ...
    + ...
10  exp(1i*new_data.TX0_RX2(r))* ...
11  conj(exp(1i*squeeze(calib_curve(RadarPosition(r)+1,2,(:, :)))) ...
    + ...
12  exp(1i*new_data.TX2_RX2(r))* ...
13  conj(exp(1i*squeeze(calib_curve(RadarPosition(r)+1,6,(:, :)))));

```

Obrázek 3.9: Násobení komplexního čísla s fází přijatého signálu s komplexním číslem s fází z kalibrační křivky.

Praktické řešení vychází z charakteru prostředí MATLAB, který je velmi dobře optimalizován k práci s velkými maticemi. Jak je vidět na obrázku 3.9 matice komplexních čísel `multi_matrix` je součet komplexně sdružených matic kalibračních křivek, vynásobených komplexním číslem s fází přijatého

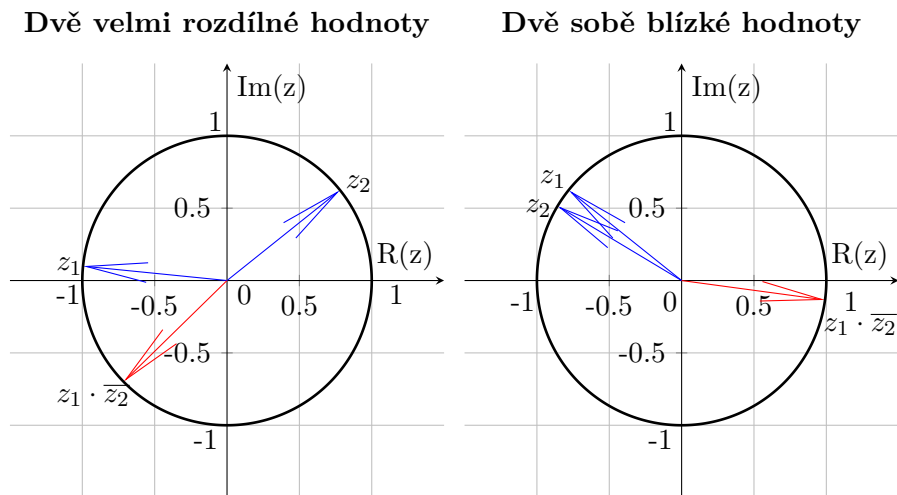
signálu. Na fáze, jak přijatého signálu, tak i v kalibračních křivkách, je použita funkce $\exp()$, protože se tím z prosté fáze signálu stane komplexní číslo,

$$e^{i\phi} = \cos(\phi) + i \sin(\phi) = a + bi, \quad (3.1)$$

kde $|a + bi| = 1$. Díky komplexnímu sdružení $\text{conj}()$ se docílí toho, že výsledkem bude komplexní číslo s fází odpovídající fázovému rozdílu násobených čísel,

$$z_1 \cdot \bar{z}_2 = e^{i\phi_1} \cdot e^{-i\phi_2} = e^{i(\phi_1 - \phi_2)}. \quad (3.2)$$

Pokud tedy bude rozdíl ve fázi malý, výsledné komplexní číslo se bude blížit jedné, jako je vidět na obrázku 3.10 vpravo. Naopak, pokud bude fázový rozdíl velký, výsledné číslo bude dále od jedné, jako na obrázku 3.10 vlevo.

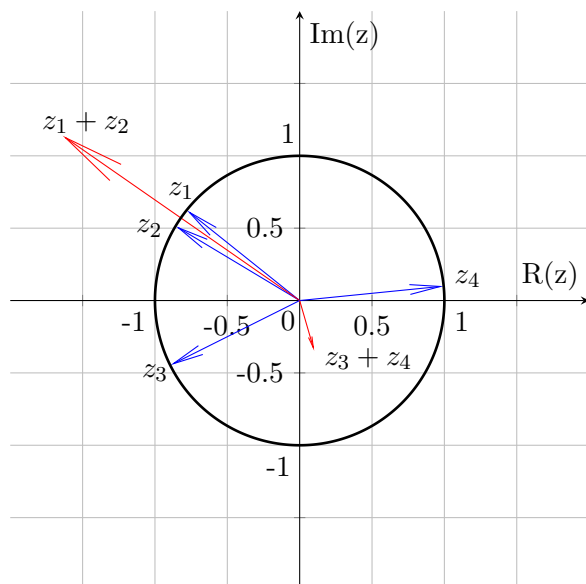


Obrázek 3.10: Ukázka násobení dvou komplexních čísel v závislosti na vzájemném fázovém posunu.

Po vynásobení matice kalibračních křivek komplexním číslem s fází přijatého signálu vznikne matice, která obsahuje komplexní hodnoty očištěné o fázové posuny, vůči referenční anténě. Následně je potřeba zjistit, pro jakou kombinaci azimutu a elevace, ze všech dvojic antén, vychází fáze, co možná nejbližší k fázi referenční antény. Sečtením matic se dosáhne toho, že největší hodnota v absolutní velikosti bude právě ta, která byla ve všech dílčích maticích fázově nejbližší k referenční anténě. A to proto, že pokud se sčítají dvě komplexní čísla se stejnou fází, pak se absolutní hodnota součtu rovná součtu jejich absolutních hodnot a fáze signálu zůstane stejná (jak je vidět na obrázku 3.11). Pokud je ale fáze rozdílná, pak je absolutní hodnota součtu menší než součet absolutních hodnot těchto dvou komplexních čísel.

Posledním krokem je tedy nalezení maxima absolutní hodnoty v matici a pozice maxima v matici, přepočtem získat azimut a elevaci vůči radaru

Součet komplexních čísel



Obrázek 3.11: Ukázka jak ovlivní rozdíl fáze komplexních čísel absolutní hodnotu jejich součtu.

(obrázek 3.12). Proměnná `indx2az_deg` je vektor od 0° do 180° s krokem $0,1^\circ$, který vychází z rozměru kalibračních křivek pro azimut. Proměnná `indx2el_deg` je také vektor, ale elevace. Tu radar určuje od 65° do 115° s krokem 5° .

Matice `multi_matrix` je vynásobena prvek po prvku s `FoV2d`, což je maska zorného pole pro příslušný radar a následně je pomocí funkce `max()` nalezeno maximum a jeho lineární index. Lineární index je pomocí funkce `ind2sub()` převeden na řádkový a sloupcový index a tyto indexy jsou dále převedeny přímo do azimutu a elevace v radiánech. Od hodnoty azimutu je odečteno $\frac{\pi}{2}$, aby nulový azimut směřoval rovně před radar.

```

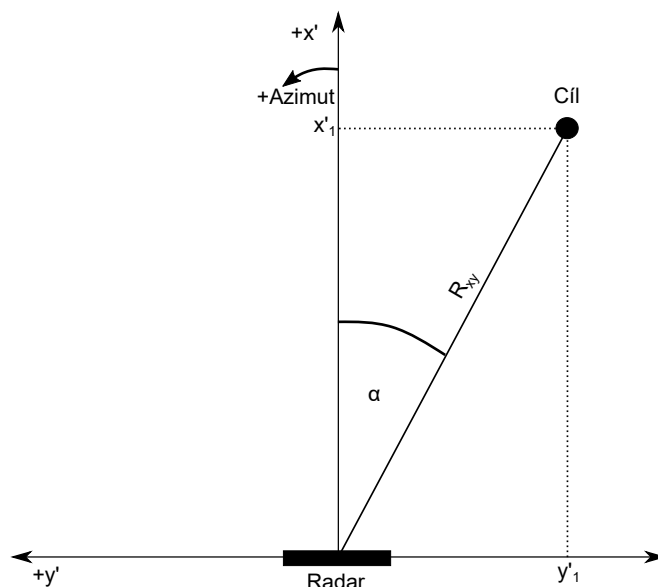
1     indx2az_deg = 0:0.1:180;
2     indx2el_deg = 65:5:115;
3     multi_matrix = ... % vynasobeni maskou FoV
4         multi_matrix.*squeeze(FoV2d(RadarPosition(r)+1, :, :));
5     [max_val, lin_indx] = ... % hledani maximalni hodnoty
6         max(multi_matrix, [], 'all', 'linear');
7     [elevation, azimuth] = ... % prevod z lin. indexu
8         ind2sub(size(multi_matrix), lin_indx);
9     azimuth = (indx2az_deg(azimuth))*(pi/180) - pi/2;
10    elevation = indx2el_deg(elevation)*(pi/180);

```

Obrázek 3.12: Nalezení maxima a převod indexu matice na azimut a elevaci.

■ Výpočet souřadnic

V této části je vysvětlen výpočet globálních souřadnic. Tedy souřadnic vůči souřadnému systému automobilu, kde počátek se nachází uprostřed zadní osy automobilu a souřadné osy jsou orientovány jako na obrázku 3.1. Postup je pro každý radar stejný. Jelikož je známa vzdálenost cíle od radaru, ne od středu zadní osy vozidla, jako první je třeba určit lokální souřadnice (obrázek 3.13) vůči radaru a ty následně transformovat do globálních souřadnic.



Obrázek 3.13: Lokální souřadnice vůči radaru. Pohled na rovinu X-Y ze směru Z.

Výpočet lokálních souřadnic vychází z obrázku 3.13, kde α je azimut cíle, R_{xy} je vzdálenost od radaru k cíli v rovině XY (podle rovnice (3.4)), x'_1 a y'_1 jsou lokální souřadnice cíle. Z toho vyplývá, že

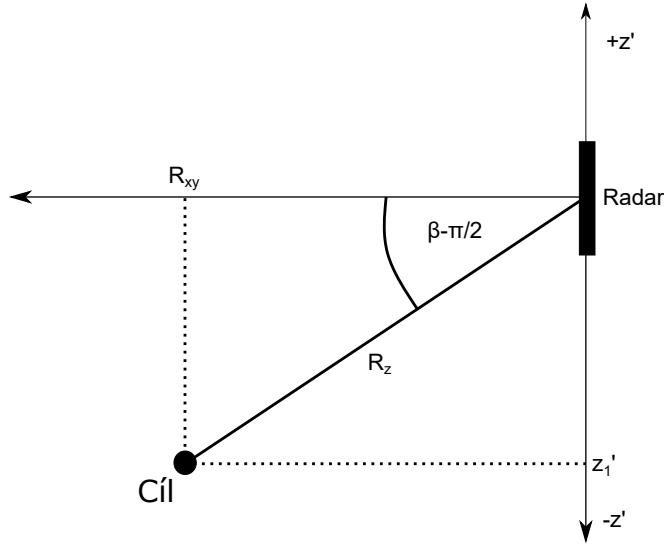
$$x'_1 = R_{xy} \cdot \cos(\alpha), \quad y'_1 = R_{xy} \cdot \sin(\alpha). \quad (3.3)$$

Výška cíle vůči radaru a vzdálenost xy_dist na obrázku 3.14, kde β je elevace cíle, R_z je vzdálenost cíle od radaru a z'_1 je výška cíle vůči radaru, se vypočítá jako,

$$z'_1 = R_z \cdot \sin\left(\beta - \frac{\pi}{2}\right), \quad xy_dist = R_z \cdot \cos\left(\beta - \frac{\pi}{2}\right). \quad (3.4)$$

Dále je nutné transformovat lokální souřadnice do souřadnic globálních. Každý radar je vůči středu globálních souřadných os posunut a jeho azimut otočen. Tyto údaje jsou k nalezení v tabulce 3.1. Jako první je potřeba provést otočení lokálních souřadných os okolo osy z zpět o úhel, o který je radar otočen. Poté je třeba provést jejich posunutí do středu zadní osy automobilu, čímž se dokončí transformace na globální souřadnice.

K otočení je použita rotační matice, která vychází ze znalostí lineární algebry a běžně se používá při různých grafických aplikacích. Existují ovšem



Obrázek 3.14: Lokální souřadnice Z vůči radaru.

dvě různé rotace. Rotace aktivní, která provádí otáčení bodu v souřadnicích a rotace pasivní, která naopak rotuje souřadné osy. V tomto případě je potřeba rotace pasivní k otočení lokálních souřadných os. Rotační matice pro pasivní rotaci se od aktivní liší tím, že jde o její inverzní matici,

$$R^{-1}(\phi) = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.5)$$

Úhel ϕ ve výrazu (3.5) je kladný pro směr otáčení proti směru hodinových ručiček. Tedy úhel, o který je třeba otočit lokální souřadnice je záporně vzatý úhel v posunu v tabulce 3.1. Konkrétně matice rotace pro radar 0, který je vůči azimutu v globálních souřadnicích posunut o 110° vypadá následovně,

$$R^{-1}(-110^\circ) = \begin{bmatrix} \cos(-110^\circ) & \sin(-110^\circ) & 0 \\ -\sin(-110^\circ) & \cos(-110^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.6)$$

Celá transformace z lokálních souřadnic do globálních vypadá takto,

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x'_1 \\ y'_1 \\ z'_1 \end{bmatrix} + \begin{bmatrix} x_{pos} \\ y_{pos} \\ z_{pos} \end{bmatrix}, \quad (3.7)$$

kde $\begin{pmatrix} x_{pos} \\ y_{pos} \\ z_{pos} \end{pmatrix}$ je vektor posunu radaru vůči středu globálních souřadnic.

Celý výpočet lokálních souřadnic na globální v kódu je ukázán na obrázku 3.15. Proměnná `xy_dist` je vzdálenost cíle v rovině XY, `loc_coord` vektor lokálních souřadnic, `glob_coord` vektor globálních souřadnic, `coodr_shift`

```

1   xy_dist = new_data.CoGRange_m(r)*cos(elevation-pi/2);
2   loc_coord = [xy_dist*cos(azimuth), xy_dist*sin(azimuth), ...
                new_data.CoGRange_m(r)*sin(elevation-pi/2)];
3   glob_coord = ... % rotace lokálních souřadnic
4   squeeze(R(new_data.RadarPosition(r)+1, :, :))*loc_coord';
5   glob_coord = glob_coord + ...
                coord_shift(new_data.RadarPosition(r)+1, :)' ;
6   % zapis souřadnic do promenne
7   new_data.Lateral(r) = glob_coord(1);
8   new_data.Longitudinal(r) = glob_coord(2);
9   new_data.Heigth(r) = glob_coord(3);

```

Obrázek 3.15: Transformace lokálních souřadnic v kódu.

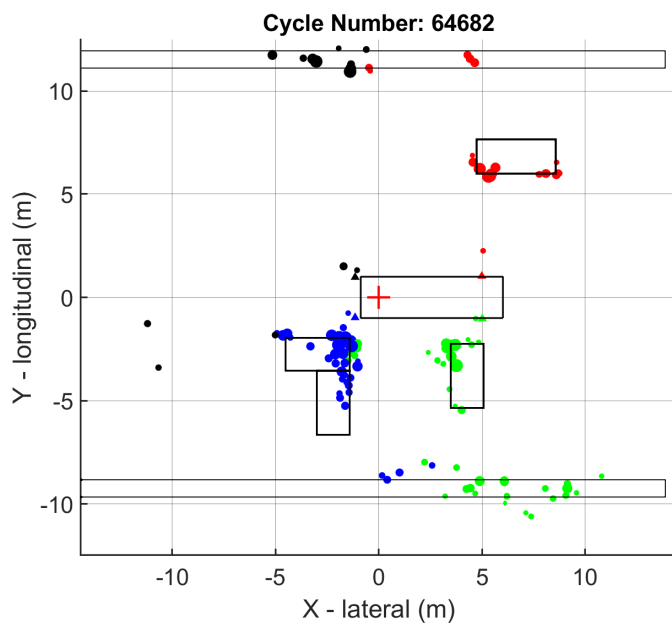
posun radaru vůči počátku a R je matice pasivní rotace. Pro každý radar musí být použita jiná rotační matice a jiný posun v souřadnicích, proto je vždy podle čísla radaru vybrána příslušná matice a vektor.

3.2.4 Vykreslení dat do 3D grafu

K interpretaci dat byl zvolen 3D graf, do kterého se postupně vykreslují cíle nalezené okolo vozidla. Díky tomu, že jsou pozice všech cílů určeny ve stejných globálních souřadnicích, je jejich vykreslování velmi jednoduché.

Na obrázku 3.23 je vidět ukázka kódu, který slouží k připravení kreslicí plochy. Nejprve je vytvořeno okno, souřadné osy a kreslicí prostředí `scatter3`. Dále jsou pojmenovány jednotlivé osy a nastaveny jejich maximální a minimální hodnoty. Proměnná `colors` obsahuje čtyři barvy (červená, zelená, modrá a černá) v RGB kódu, kterými se obarví jednotlivé cíle podle toho, jakým radarem byly detekovány. Poté je pomocí funkce `line()` vyznačen střed globálních souřadnic, jednoduše naznačený obrys automobilu a pomocí barevných trojúhelníků (opět podle barvy odpovídající detekovaným cílům) naznačené pozice radarů na vozidle. Druhý cyklus `for` slouží k postupnému vykreslení detekovaných cílů. Smyčka se opakuje po jednotlivých cyklech. Nejprve se vyberou všechny detekce se stejným `CycleNumber` a poté se vykreslí jejich pozice v souřadnicích pomocí funkce `set()`. Barva zobrazených cílů závisí na radaru, kterým byly detekovány a velikost je dána podle SNR detekce. Následuje vypsání kolikátý cyklus se vykreslil a čeká se 50 ms, protože právě takovou dobu trvá jeden cyklus.

Na obrázku 3.17 je ukázka z videa pořízeného z kabiny řidiče vozidla při testovací jízdě. Tento záběr je z druhé poloviny jízdy, kdy automobil bude vpravo míjet dvě zaparkovaná vozidla (jedno šedé a druhé modré barvy). Dále za vozidly se nachází betonová zídka. Vlevo od vozidla se nachází zeď budovy a další zaparkované auto černé barvy. Na obrázku 3.16 je pohled na rovinu XY (pohled z ptáčích perspektivy). Tento záběr není ze stejného okamžiku jako obrázek 3.17, protože radary jsou nasměrovány tak, že koukají šikmo ve směru za vozidlo, takže objekty, které jsou před vozidlem se v grafu radarů nezobrazují.



Obrázek 3.16: 3D graf zobrazující detekované cíle.



Obrázek 3.17: Náhled z kabiny vozidla na vnější prostředí. Obrázek je vyjmutý z dodaného videa z testovací jízdy.

Z detekovaných cílů na obrázku 3.16 lze identifikovat všechna tři vozidla

v okolí automobilu. Zejména radar číslo dva (v modré barvě) detekuje velké množství cílů na šedém vozidle. Zřetelný je obrys vozidla a další cíle jsou nejspíše odrazy od kapoty a předního skla. Dále je možné rozpoznat obrys modrého vozidla, detekovaného radarem číslo jedna (v zelené barvě). Oba tyto radary detekují ve vzdálenosti okolo 10 metrů betonovou zídku. V grafu jsou také rozpoznatelné odrazy způsobené od černého vozidla (na obrázku 3.17 vlevo) z radaru číslo nula (v červené barvě). V horní části grafu jsou také vidět dorazy od zdi budovy vzdálené asi 11 metrů.

■ 3.2.5 Dosažené výsledky

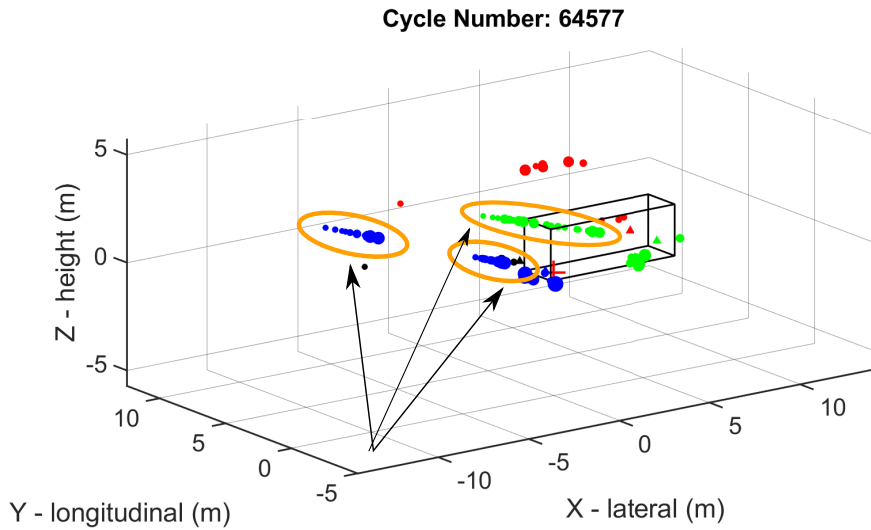
Při zpracovávání dat v praktické části práce se vyskytly dva hlavní problémy. Oba se objevili až při bližším zkoumání pozic výsledných cílů. Díky tomu, že v dodaných datech byla k dispozici informace o souřadnicích cíle, bylo možné porovnat výsledky vlastních výpočtů s výpočty z firmy Valeo autoklimatizace, k.s.

■ Komplikace s nulovými fázory

Prvním problémem bylo, že se v některých cyklech detekované cíle zobrazovaly na přímce vedené od radaru směrem za vozidlo. Tyto obrazce se objevovaly zejména u radarů číslo jedna a dva v náhodných cyklech a vždy po krátké době zmizely a cíle se vykreslovaly bez problémů. Na obrázku 3.18 jsou vidět právě tyto podezřelé cíle. V těchto místech se v prostoru nic nenachází, zelené cíle jsou dokonce uvnitř vozidla a v praxi není možné, aby se detekce vyskytovaly takto hustě vysázené za sebou, bez jiných cílů v okolí. Je jasné, že se nemůže jednat o skutečné objekty. Proto bylo důležité identifikovat, čím je tato chyba způsobena.

První možností mohla být chyba ve výpočtu souřadnic, což by ale znamenalo, že by musely být i ostatní cíle zobrazeny na špatných místech. Takový problém nenastával, ostatní cíle byly zobrazeny na smysluplných místech. Druhou možností bylo přetečení paměti některé proměnné, což bylo opět hlubším zkoumáním kódu vyloučeno. Další chybou mohlo být nesprávné určování azimutu a elevace, nebo násobení matice `multi_matrix` a vyhodnocování maxima. Opět kontrolou kódu byla tato chyba vyloučena a dalším důvodem, proč zde nemohl být problém bylo, že by tím byly ovlivněny i ostatní výsledky. Poslední možností tedy bylo, že v samotných zpracovávaných datech se vyskytovala nějaká vada. Při konkrétním zkoumání komplexních hodnot bylo zjištěno, že v inkriminovaných cyklech se v komplexních hodnotách fázorů z jednotlivých antén vyskytují samé nulové hodnoty (obrázek 3.19). Zde je na první pohled jasné, že nejde o reálná data. Při správné funkci by se nikdy nemohlo stát, že by komplexní hodnota vyšla nulová, a že by fázový posun každé dvojice antén od referenční dvojice byl nulový.

Závěrem je, že tato data nejsou validní a nemohou být použita k vyhodnocování azimutu a elevace, tudíž tyto detekce nebyly uvažovány. Nejspíše došlo při exportování komplexních dat z radaru k chybě. Rozhodně se nejedná o chybu v radarech, či při přenosu signálů z antén do procesoru radaru,



Obrázek 3.18: Ukázka podezřelých cílů v 3D grafu. V obrázku nejsou zobrazeny informace o souřadnicích, rychlosti, azimutu, atd., protože pro identifikaci vadných dat nejsou podstatné.

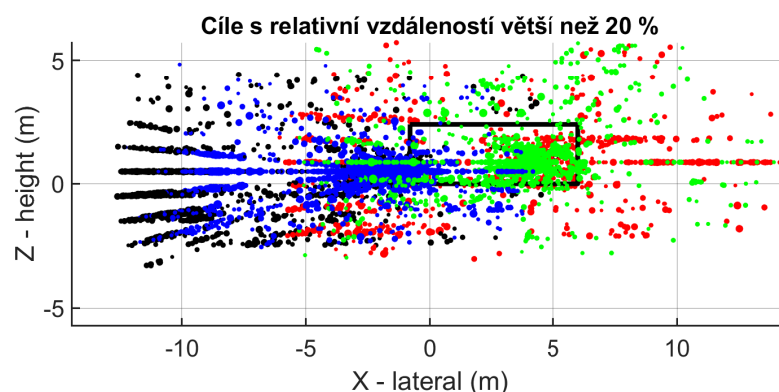
protože i tyto detekce mají spočítanou pozici vůči automobilu. K chybě tedy muselo dojít až po signálovém zpracování radarem.

	1	2	3	4	5	6	7	8	9
	CycleNumber	TX2_RX0_real	TX2_RX0_imag	TX1_0_RX0_real	TX1_0_RX0_imag	TX1_180_RX0_real	TX1_180_RX0_imag	TX2_RX1_real	TX2_RX1_imag
31015	64577	11	-25	1	-19	-7	-20	-17	20
31016	64577	-7	15	-7	9	-6	5	35	-2
31017	64577	-57	75	52	-4	50	-10	-65	24
31018	64577	-22	-124	-33	36	-39	35	19	-98
31019	64577	-99	60	36	-31	32	-24	-62	67
31020	64577	27	-95	-16	47	-20	45	-3	-71
31021	64577	0	0	0	0	0	0	0	0
31022	64577	0	0	0	0	0	0	0	0
31023	64577	0	0	0	0	0	0	0	0
31024	64577	0	0	0	0	0	0	0	0
31025	64577	0	0	0	0	0	0	0	0
31026	64577	0	0	0	0	0	0	0	0
31027	64577	0	0	0	0	0	0	0	0
31028	64577	0	0	0	0	0	0	0	0
31029	64577	0	0	0	0	0	0	0	0
31030	64577	0	0	0	0	0	0	0	0
31031	64577	0	0	0	0	0	0	0	0
31032	64577	0	0	0	0	0	0	0	0
31033	64577	0	0	0	0	0	0	0	0
31034	64577	0	0	0	0	0	0	0	0
31035	64577	0	0	0	0	0	0	0	0
31036	64577	0	0	0	0	0	0	0	0
31037	64577	0	0	0	0	0	0	0	0

Obrázek 3.19: Vadná data způsobující chybné výsledky.

■ Komplikace se zaměněným pořadím cyklů

Druhým problémem bylo, že v některých cyklech, na první pohled naprosto náhodných, byla část cílů zobrazena na úplně jiném místě. Při přehrávání celého záznamu 3D grafu se objevovali místa, kdy cíle na pár cyklů změnily



Obrázek 3.20: Cíle, jejichž relativní vzdálenost od cílů podle Valeo autoklimatizace, k.s. je větší než 20 % z pohledu na rovinu XZ.

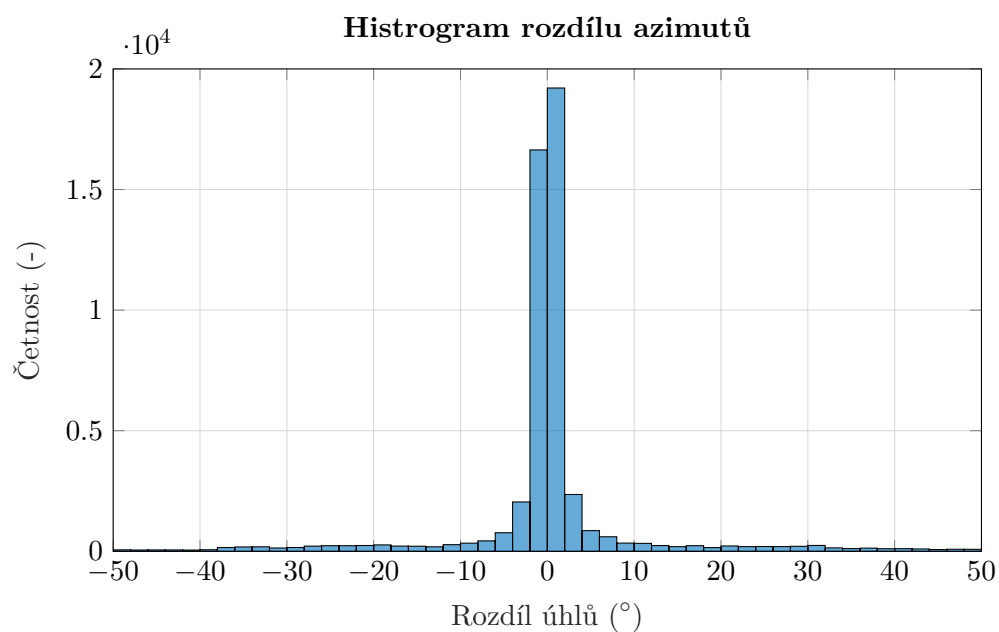
svou pozici a poté se vrátili zpět do původní polohy. Tyto podezřelé pozice se zejména objevovali v místě za vozidlem, kde se žádný objekt nevyskytoval. Cíle, jejichž pozice se liší od pozic určených systémem ve firmě Valeo autoklimatizace, k.s. v relativní vzdálenosti (vzdálenost mezi pozicemi vydělená vzdáleností od vozidla) o více než 20 %, jsou zobrazeny v obrázku 3.20. Zde je zřetelné, že se cíle opět objevují na přímkách odpovídající stejným elevacím (v tomto obrázku nejsou zahrnuta vadná data, která byla popsána v sekci 3.2.5). Toto chování znovu ukazuje na nějakou chybu. Opět byl proveden stejný sled kroků jako při odhalování problému v sekci 3.2.5 a žádná chyba v kódu nebyla zjištěna. Po důkladnější analýze samotných dat byla opět nalezena vada. Na obrázku 3.21 jsou ukázané detekce, jejichž pozice je nesprávně určena. Je zde vidět, že data jdou za sebou podle čísla cyklu 64571, poté se objeví 12 detekcí s číslem cyklu 64572 a následně pokračují detekce cyklu 64571. Takových míst, kde se detekce předbíhají je v datech několik a pokaždé se objevují tam, kde jsou cíle nesprávně vykreslené. Navíc ve většině případů se po tomto přehození také objevují detekce s nulovými fázory. To opět jasně ukazuje na chybu při exportu dat.

■ Shrnutí výsledků

Po vyjasnění problémů, které se při zpracování dat vyskytly, lze říci, že vypracovaný skript v prostředí MATLAB je plně funkční. Z validních fázorů je možné správně určit azimut a elevaci, následně i pozice cíle v globálních souřadnicích. Na obrázku 3.22 je znázorněn histogram rozdílů azimutů určených vlastním výpočtem a zpracováním ve firmě Valeo autoklimatizace, k.s. Je vidět, že data vykazují charakter normálního rozdělení, což znamená, že velká část azimutů je určena s malou odchylkou. Jelikož vadné vzorky fázorů, které vznikly chybou při exportu nelze přesně identifikovat a tím pádem odfiltrovat, je tento histogram ovlivněn cíli, které se nedají považovat za validní.

	1	2	3	4	5	6	7	8	9	10
	CycleNumber	RadarPosition	CoGDoppler_mps	CoGRange_m	SNR_half_dB	Azimuth_rad	Elevation_rad	Lateral_m	Longitudinal_m	Heighth_m
30362	64571	2	0.9492	10.5000	31	-2.1119	1.6022	-6.5592	-9.9718	0.8198
30363	64571	2	1.0898	10.6797	29	-2.2934	1.4870	-8.1918	-8.9589	-0.4037
30364	64571	2	1.0977	10.9063	28	-2.2724	1.5691	-8.1935	-9.3061	0.4710
30365	64571	2	1.1367	11.1719	29	-2.3422	1.5970	-8.9400	-8.9825	0.7824
30366	64572	0	0	0.0820	113	3.4907	1.5708	4.8999	1.0019	0.8600
30367	64572	0	0.1719	1.0898	32	1.8518	1.7506	4.6796	2.0602	1.0549
30368	64572	0	0.4141	1.1719	30	1.0489	1.5289	5.5607	2.0450	0.8109
30369	64572	0	0.8164	1.2695	35	2.4173	1.9042	4.0785	1.8249	1.2754
30370	64572	0	-0.5273	1.2656	29	1.3195	1.8309	5.2812	2.2146	1.1854
30371	64572	0	-0.0039	1.3047	33	1.4835	1.8361	5.0867	2.2843	1.2021
30372	64572	0	-0.7305	1.4063	28	3.0543	1.2235	3.6598	1.1452	0.3813
30373	64572	0	0.9453	1.4414	29	3.0980	1.1415	3.6677	1.0872	0.2600
30374	64572	0	-0.2148	1.4414	29	1.1223	1.9810	5.5502	2.2211	1.4348
30375	64572	0	0.7383	10.7578	27	2.0176	1.5935	0.3299	10.7292	1.1041
30376	64572	0	0.7422	10.9023	34	2.1188	1.5202	-0.6960	10.3238	0.3084
30377	64572	0	-0.5664	11.1367	54	1.2776	1.4940	8.1864	11.6600	0.0056
30378	64571	3	0	0.0820	118	3.9270	1.5708	-1.2100	0.9180	0.4900
30379	64571	3	0	0.3555	35	1.4731	1.4190	-1.1177	1.3257	0.4362
30380	64571	3	0.0039	0.5039	33	0.7854	1.3963	-0.8011	1.3269	0.4025
30381	64571	3	0.8789	0.8008	45	2.1974	1.9338	-1.5910	1.5824	0.7744

Obrázek 3.21: Ukázka dat, která mají zaměněné pořadí řádků.



Obrázek 3.22: Histogram rozdílů azimutů vypočtených a určených podle Valeo autoklimatizace, k.s.

```

1   hFig = figure;
2   hAx = axes(hFig);
3   hScatter = scatter3(hAx, nan, nan, nan);
4   hTitle = title(hAx, '');
5   xlabel('X - lateral (m)')
6   ylabel('Y - longitudinal (m)')
7   zlabel('Z - height (m)')
8   view(hAx, -75, 25);
9   maxX = max(abs(new_data.Lateral));
10  maxY = max(abs(new_data.Longitudinal));
11  maxZ = max(abs(new_data.Heigth));
12  axis(hAx, 'equal');
13  xlim(hAx, [-maxX, maxX])
14  ylim(hAx, [-maxY, maxY])
15  zlim(hAx, [-maxZ, maxZ])
16  colors = [1 0 0; 0 1 0; 0 0 1; 0 0 0];
17  hPocatek = line(hAx, 'XData', 0, 'YData', 0, 'ZData', 0, ...
18    'LineWidth', 2, ...
19    'Marker', '+', 'MarkerSize', 20, 'Color', 'r');
20  car = line(hAx, 'XData', [6 6 -0.8 -0.8 6 6 6 -0.8 -0.8 ...
21    6 nan 6 6 nan -0.8 -0.8 nan -0.8 -0.8], ...
22    'YData', [-1 1 1 -1 -1 -1 1 1 -1 -1 nan 1 1 nan 1 1 ...
23    nan -1 -1], ...
24    'ZData', [0 0 0 0 0 2.39 2.39 2.39 2.39 2.39 nan 0 ...
25    2.39 nan 0 2.39 nan 0 2.39], ...
26    'LineWidth', 1, 'LineStyle', '-', 'MarkerSize', 50, ...
27    'Color', 'k');
28  for i = 1:4
29    line(hAx, 'XData', coord_shift(i,1), 'YData', ...
30      coord_shift(i,2), 'ZData', ...
31      coord_shift(i,3), 'Marker', '^', 'LineStyle', 'none', ...
32      'Color', colors(i,:), 'MarkerFaceColor', colors(i,:));
33  end
34  [minCycle, maxCycle] = bounds(new_data.CycleNumber);
35  for iCycle = minCycle:maxCycle
36    logIndCycle = new_data.CycleNumber == iCycle;
37    radInd = new_data.RadarPosition(logIndCycle);
38    sizes = rescale(new_data.SNR_half_dB(logIndCycle), 5, 40);
39    set(hScatter, 'XData', new_data.Lateral(logIndCycle), ...
40      'YData', new_data.Longitudinal(logIndCycle), ...
41      'ZData', new_data.Heigth(logIndCycle), ...
42      'SizeData', sizes, ...
43      'CData', colors(radInd+1, :), ...
44      'MarkerFaceColor', 'flat');
45    hTitle.String = sprintf('Cycle Number: %d', iCycle);
46    pause(0.05)
47  end

```

Obrázek 3.23: Příprava kreslicí plochy grafu a cyklus postupného vykreslení dat.

Kapitola 4

Závěr

V práci byly popsány základy analýzy signálu MIMO FMCW radaru. V teoretické části bylo rozebráno na jakém principu radary tohoto typu pracují, jakým způsobem se určuje vzdálenost, rychlost, azimut a elevace cíle a jaké existují různé přístupy z hlediska signálového zpracování.

V praktické části byla popsána testovací jízda vozidla, na kterém byly umístěny čtyři MIMO FMCW radary. Naměřená data z této jízdy byla podrobně popsána a následně použita ke zpracování v prostředí Mathworks MATLAB. Také zde byly zmíněny charakteristiky radarů jako např. FoV nebo kalibrační křivky. Poté bylo ukázáno, jak je možné určit azimut a elevaci ze změřených fázorů na elementech virtuálního anténního pole pomocí kalibračních dat. Následně byly zjištěny souřadnice cílů a byly vykresleny do 3D grafu, ze kterého je možné pomocí dostupného videa z testovací jízdy určit objekty v okolí vozidla. Bohužel komplexní data byla zřejmě při exportu poškozena a některé fázory neodpovídají skutečným detekcím, a tím jsou výsledky zkreslené. Z dat nebylo možné přesně určit, které části nejsou validní.

Vhodným řešením by bylo testovací jízdu opakovat a zkontrolovat správnost exportovaných dat. Poté by bylo možné objektivně zhodnotit výsledky zpracování a případně kód upravit tak, aby byly se dosahovalo přesnějšího určení pozice cílů.



Seznam zkratek

- ADC** Analog to digital convertor
- CFAR** Constant false alarm rate
- CSV** Comma-separated values
- DFT** Discrete Fourier transform
- ESPRIT** Estimation of signal parameters by rotational invariance techniques
- FFT** Fast Fourier transform
- FMCW** Frequency modulated continuous wave
- FoV** Field of view
- LNA** Low noise amplifier
- LPF** Low pass filter
- MIMO** Multiple input multiple output
- MUSIC** Multiple signal classification
- PA** Power amplifier
- PD** Power divider
- RCS** Radar cross section
- RX** Receive
- SIMO** Single input single output
- SISO** Single input single output
- SNR** Signal to noise ratio
- TX** Transmission
- VCO** Voltage controlled oscillator



Literatura

- [1] ROOS, Fabian, Jonathan BECHTER, Christina KNILL, Benedikt SCHWEIZER a Christian WALDSCHMIDT. Radar Sensors for Autonomous Driving: Modulation Schemes and Interference Mitigation. *IEEE Microwave Magazine* [online]. 2019, **20**(9), 58-72 [cit. 2022-01-16]. ISSN 1527-3342. Dostupné z: doi:10.1109/MMM.2019.2922120
- [2] HASCH, Jürgen, Eray TOPAK, Raik SCHNABEL, Thomas ZWICK, Robert WEIGEL a Christian WALDSCHMIDT. Millimeter-Wave Technology for Automotive Radar Sensors in the 77 GHz Frequency Band. *IEEE Transactions on Microwave Theory and Techniques* [online]. 2012, **60**(3), 845-860 [cit. 2022-01-10]. ISSN 0018-9480. Dostupné z: doi:10.1109/TMTT.2011.2178427
- [3] BROOKER, Graham. Understanding Millimetre Wave FMCW Radars. In: *1st International Conference on Sensing Technology*. Palmerston North, New Zeland, 2005, s. 152-157.
- [4] SULEYMANOV, Suleyman. *Design and implementation of an FMCW radar signal processing module for automotive applications*. Enschede, 2016. Diplomová práce. University of Twente.
- [5] KRONAUGE, Matthias, Christoph SCHROEDER a Hermann ROHLING. Radar target detection and Doppler ambiguity resolution. In: *11-th INTERNATIONAL RADAR SYMPOSIUM*. Vilnius: IEEE, 2010, s. 1-4. ISBN 978-1-4244-5614-7. ISSN 2155-5753.
- [6] RAO, Sandeep Keshava. *MIMO radar*. Texas Instruments. 2018. Dostupné také z: <https://www.ti.com/lit/pdf/swra554?keyMatch=SWRA554A>
- [7] KNOTT, Eugene F., John F. SHAEFFER a Michael T. TULEY. *Radar Cross Section*. Second Edition. Raleigh: SciTech Publishing, 2004. ISBN 1-891121-1.
- [8] QUINQUIS, A., E. RADOI a F.-C. TOTIR. Some Radar Imagery Results Using Superresolution Techniques. *IEEE Transactions on Antennas and Propagation* [online]. 2004, **52**(5), 1230-1244 [cit. 2022-01-11]. ISSN 0018-926X. Dostupné z: doi:10.1109/TAP.2004.827541

- [9] CAO, Tri-Tan V., James PALMER a Paul E. BERRY. False alarm control of CFAR algorithms with experimental bistatic radar data. *2010 IEEE Radar Conference* [online]. IEEE, 2010, 2010, , 156-161 [cit. 2022-01-15]. ISBN 978-1-4244-5811-0. Dostupné z: doi:10.1109/RADAR.2010.5494633
- [10] SKOLNIK, Merrill I. *Introduction to radar systems*. 2. ed. Tokyo: MacGraw-Hill, 1980. ISBN 00-705-7909-1.

Přílohy

```
1      % Zpracovani dat z automobiloveho radaru
2      clear;
3
4      %% Nacteni dat
5
6      raw_data = readtable('RoCList.csv');
7
8      %% Filtrace dat
9
10     % spodni mez filtrace podle SNR
11     low_SNR = 30;
12     % horni mez filtrace podle SNR
13     high_SNR = 100;
14     % cyklus od ktereho se zacinaji zpracovavat data
15     first_cycle = 64302;
16
17     sorted_data = raw_data(raw_data.SNR_half_dB > low_SNR ...
18     & raw_data.SNR_half_dB < high_SNR, :);
19     sorted_data = sorted_data(sorted_data.CycleNumber ≥ ...
20     first_cycle, :);
21
22     %% Odstraneni nepotrebnych dat a prejmenovani promennych
23
24     sorted_data.Infra = [];
25     sorted_data.NonInfra = [];
26     sorted_data.TX1_180_RX0_real = [];
27     sorted_data.TX1_180_RX0_imag = [];
28     sorted_data.TX1_180_RX1_real = [];
29     sorted_data.TX1_180_RX1_imag = [];
30     sorted_data.TX1_180_RX2_real = [];
31     sorted_data.TX1_180_RX2_imag = [];
32
33     % jen formalni preznaceni anten: TX2 -> TX0, TX1 -> TX2
34     sorted_data.Properties.VariableNames{'TX2_RX0_real'} = ...
35     'TX0_RX0_real';
36     sorted_data.Properties.VariableNames{'TX2_RX0_imag'} = ...
37     'TX0_RX0_imag';
38     sorted_data.Properties.VariableNames{'TX1_0_RX0_real'} = ...
39     'TX2_RX0_real';
40     sorted_data.Properties.VariableNames{'TX1_0_RX0_imag'} = ...
41     'TX2_RX0_imag';
```

```

37     sorted_data.Properties.VariableNames{'TX2_RX1_real'} = ...
        'TX0_RX1_real';
38     sorted_data.Properties.VariableNames{'TX2_RX1_imag'} = ...
        'TX0_RX1_imag';
39     sorted_data.Properties.VariableNames{'TX1_0_RX1_real'} = ...
        'TX2_RX1_real';
40     sorted_data.Properties.VariableNames{'TX1_0_RX1_imag'} = ...
        'TX2_RX1_imag';
41     sorted_data.Properties.VariableNames{'TX2_RX2_real'} = ...
        'TX0_RX2_real';
42     sorted_data.Properties.VariableNames{'TX2_RX2_imag'} = ...
        'TX0_RX2_imag';
43     sorted_data.Properties.VariableNames{'TX1_0_RX2_real'} = ...
        'TX2_RX2_real';
44     sorted_data.Properties.VariableNames{'TX1_0_RX2_imag'} = ...
        'TX2_RX2_imag';
45
46     %% priprava nove tabulky
47
48     rows = size(sorted_data, 1);    % pocet radku v matici dat
49     new_data = sorted_data(:,1:5);
50
51     new_data.Azimuth = zeros(rows,1);    % azimut cile
52     new_data.Elevation = zeros(rows,1); % elevace v radianech
53     new_data.Lateral = zeros(rows,1);   % souradnice cile X
54     new_data.Longitudinal = zeros(rows,1); % souradnice cile Y
55     new_data.Heigth = zeros(rows,1);    % souradnice cile Z
56
57     new_data.TX0_RX0 = zeros(rows,1);
58     new_data.TX2_RX0 = zeros(rows,1);
59     new_data.TX0_RX1 = zeros(rows,1);
60     new_data.TX2_RX1 = zeros(rows,1);
61     new_data.TX0_RX2 = zeros(rows,1);
62
63     %% Nacteni kalibracnich krivek
64     calib_curve = zeros(4,7,11,1801);
65     FoV2d_37 = zeros(4,37,37);
66     FoV2d = zeros(4,11,1801);
67
68     load dataHeaderFile_MB79_MB79_NF_43.mat;
69     calib_curve(1,:,:,:) = ...
        dataHeaderFile.Beam1.phaseDiff2dSpdo_full;
70     FoV2d_37(1,:,:,:) = dataHeaderFile.Beam1.FoV2d;
71
72     load dataHeaderFile_MB79_MB79_NF_44.mat;
73     calib_curve(2,:,:,:) = ...
        dataHeaderFile.Beam1.phaseDiff2dSpdo_full;
74     FoV2d_37(2,:,:,:) = dataHeaderFile.Beam1.FoV2d;
75
76     load dataHeaderFile_MB79_MB79_NF_42.mat;
77     calib_curve(3,:,:,:) = ...
        dataHeaderFile.Beam1.phaseDiff2dSpdo_full;
78     FoV2d_37(3,:,:,:) = dataHeaderFile.Beam1.FoV2d;
79
80     load dataHeaderFile_MB79_MB79_NF_41.mat;
81     calib_curve(4,:,:,:) = ...
        dataHeaderFile.Beam1.phaseDiff2dSpdo_full;

```

```

82     FoV2d_37(4, :, :) = dataHeaderFile.Beam1.FoV2d;
83
84     %% interpolace FoV matic pro kazdy radar
85
86     [X, Y] = meshgrid(linspace(1,37,1801), linspace(1, 11, 11));
87
88     FoV2d(1, :, :) = interp2(squeeze(FoV2d_37(1,14:24,:)), X, ...
89                             Y, 'nearest');
90     FoV2d(2, :, :) = interp2(squeeze(FoV2d_37(2,14:24,:)), X, ...
91                             Y, 'nearest');
92     FoV2d(3, :, :) = interp2(squeeze(FoV2d_37(3,14:24,:)), X, ...
93                             Y, 'nearest');
94     FoV2d(4, :, :) = interp2(squeeze(FoV2d_37(4,14:24,:)), X, ...
95                             Y, 'nearest');
96
97     %% Vypocet azimutu a elevace
98
99     indx_to_az_deg = 0:0.1:180;
100    indx_to_el_deg = 65:5:115;
101    azimuth = 0;
102    multi_matrix = zeros(11,1801);
103    loc_coord = [0 0 0];
104    glob_coord = [0 0 0];
105    xy_dist = 0;
106
107    %% posun radaru
108    coord_shift = zeros(4,3);
109    %           x           y           z
110    coord_shift(1,:) = [4977.5 1029.7 865]/1000;
111    coord_shift(2,:) = [5000.2 -1029.9 870]/1000;
112    coord_shift(3,:) = [-1154 -976.06 495]/1000;
113    coord_shift(4,:) = [-1152.5 976.06 495]/1000;
114
115    radar_az_shift = zeros(4);
116    radar_az_shift(1) = 110*pi/180;
117    radar_az_shift(2) = -110*pi/180;
118    radar_az_shift(3) = -135*pi/180;
119    radar_az_shift(4) = 135*pi/180;
120
121    %% matice rotace
122    R = zeros(4, 3, 3);
123
124    R(1, :, :) = [cos(-radar_az_shift(1)), ...
125                sin(-radar_az_shift(1)), 0; ...
126                -sin(-radar_az_shift(1)), cos(-radar_az_shift(1)), ...
127                0; 0, 0, 1];
128
129    R(2, :, :) = [cos(-radar_az_shift(2)), ...
130                sin(-radar_az_shift(2)), 0; ...
131                -sin(-radar_az_shift(2)), cos(-radar_az_shift(2)), ...
132                0; 0, 0, 1];
133
134    R(3, :, :) = [cos(-radar_az_shift(3)), ...
135                sin(-radar_az_shift(3)), 0; ...
136                -sin(-radar_az_shift(3)), cos(-radar_az_shift(3)), ...
137                0; 0, 0, 1];
138
139    R(4, :, :) = [cos(-radar_az_shift(4)), ...
140                sin(-radar_az_shift(4)), 0; ...

```

```

        -sin(-radar_az_shift(4)), cos(-radar_az_shift(4)), ...
        0; 0, 0, 1];
125
126 %% zpracovani dat
127 for r = 1:rows
128
129 % vyjmuti nulovych fazoru
130 if(sum(sorted_data{r,13:end})==0)
131 new_data.Azimuth(r) = nan;
132 new_data.Elevation(r) = nan;
133 new_data.Lateral(r) = nan;
134 new_data.Longitudinal(r) = nan;
135 new_data.Heigth(r) = nan;
136 continue
137 end
138
139 % faze dvojic anten
140 new_data.TX0_RX0(r) = angle(sorted_data.TX0_RX0_real(r) ...
    + 1i*sorted_data.TX0_RX0_imag(r));
141 new_data.TX2_RX0(r) = angle(sorted_data.TX2_RX0_real(r) ...
    + 1i*sorted_data.TX2_RX0_imag(r));
142 new_data.TX0_RX1(r) = angle(sorted_data.TX0_RX1_real(r) ...
    + 1i*sorted_data.TX0_RX1_imag(r));
143 new_data.TX2_RX1(r) = angle(sorted_data.TX2_RX1_real(r) ...
    + 1i*sorted_data.TX2_RX1_imag(r));
144 new_data.TX0_RX2(r) = angle(sorted_data.TX0_RX2_real(r) ...
    + 1i*sorted_data.TX0_RX2_imag(r));
145 new_data.TX2_RX2(r) = angle(sorted_data.TX2_RX2_real(r) ...
    + 1i*sorted_data.TX2_RX2_imag(r));
146
147 % komplexne sdruzene nasobeni faze s kalibracni krivkou
148 multi_matrix = exp(1i*new_data.TX0_RX0(r))* ...
    conj(exp(1i*zeros(1,1801))) + ...
149 exp(1i*new_data.TX2_RX0(r))*conj(exp(1i*squeeze ...
    (calib_curve(new_data.RadarPosition(r)+1,4,(:, :)))) + ...
150 exp(1i*new_data.TX0_RX1(r))*conj(exp(1i*squeeze ...
    (calib_curve(new_data.RadarPosition(r)+1,1,(:, :)))) + ...
151 exp(1i*new_data.TX2_RX1(r))*conj(exp(1i*squeeze ...
    (calib_curve(new_data.RadarPosition(r)+1,5,(:, :)))) + ...
152 exp(1i*new_data.TX0_RX2(r))*conj(exp(1i*squeeze ...
    (calib_curve(new_data.RadarPosition(r)+1,2,(:, :)))) + ...
153 exp(1i*new_data.TX2_RX2(r))*conj(exp(1i*squeeze ...
    (calib_curve(new_data.RadarPosition(r)+1,6,(:, :)))));
154
155 % nasobeni maskou FoV
156 multi_matrix = multi_matrix.* ...
    squeeze(FoV2d(new_data.RadarPosition(r)+1,(:, :)));
157 % hledani maxima
158 [max_val, lin_idx] = max(multi_matrix, [], 'all', ...
    'linear');
159
160 [elevation, azimuth] = ind2sub(size(multi_matrix), ...
    lin_idx);
161 azimuth = (indx_to_az_deg(azimuth))*(pi/180) - pi/2;
162 elevation = indx_to_el_deg(elevation)*(pi/180);
163
164 % vypocet souradnic vuci radaru

```



```

165 xy_dist = new_data.CoGRange_m(r)*cos(elevation-pi/2);
166 loc_coord = [xy_dist*cos(azimuth), xy_dist*sin(azimuth), ...
               new_data.CoGRange_m(r)*sin(elevation-pi/2)];
167
168 % transformace souradnic
169 glob_coord = ...
               squeeze(R(new_data.RadarPosition(r)+1, :, :))*loc_coord';
170 glob_coord = glob_coord + ...
               coord_shift(new_data.RadarPosition(r)+1, :);
171
172 new_data.Lateral(r) = glob_coord(1);
173 new_data.Longitudinal(r) = glob_coord(2);
174 new_data.Heigth(r) = glob_coord(3);
175
176 % prepocet azimutu vuci radaru
177 new_data.Azimuth(r) = ...
               -acot(new_data.Longitudinal(r)/new_data.Lateral(r))+ ...
               sign(new_data.Longitudinal(r))*pi/2;
178
179 new_data.Elevation(r) = elevation;
180 end
181
182 %% priprava na 3d obrazek
183 hFig = figure;
184 hAx = axes(hFig);
185 hScatter = scatter3(hAx, nan, nan, nan);
186 hTitle = title(hAx, '');
187 xlabel('X - lateral (m)')
188 ylabel('Y - longitudinal (m)')
189 zlabel('Z - height (m)')
190 view(hAx, -75, 25);
191 maxX = max(abs(new_data.Lateral));
192 maxY = max(abs(new_data.Longitudinal));
193 maxZ = max(abs(new_data.Heigth));
194 axis(hAx, 'equal');
195 xlim(hAx, [-maxX, maxX])
196 ylim(hAx, [-maxY, maxY])
197 zlim(hAx, [-maxZ, maxZ])
198 colors = [1 0 0; 0 1 0; 0 0 1; 0 0 0];
199 % vyznaceni pocatku
200 hPocatek = line(hAx, 'XData', 0, 'YData', 0, 'ZData', 0, ...
                  'LineWidth', 2, ...
                  'Marker', '+', 'MarkerSize', 20, 'Color', 'r');
201 % naznaceni obrysu vozidla
202 car = line(hAx, 'XData', [6 6 -0.88 -0.88 6 6 6 -0.88 ...
                          -0.88 6 nan 6 6 nan -0.88 -0.88 nan -0.88 -0.88], ...
            'YData', [-1 1 1 -1 -1 -1 1 1 -1 -1 nan 1 1 nan 1 1 ...
                      nan -1 -1], ...
            'ZData', [0 0 0 0 0 2.39 2.39 2.39 2.39 2.39 nan 0 ...
                      2.39 nan 0 2.39 nan 0 2.39], ...
            'LineWidth', 1, 'LineStyle', '-', 'MarkerSize', 50, ...
            'Color', 'k');
203
204 % vyznaceni polohy radaru
205 for i = 1:4
206     line(hAx, 'XData', coord_shift(i,1), 'YData', ...
          coord_shift(i,2), 'ZData', ...
          coord_shift(i,3), 'Marker', '^', 'LineStyle', 'none', ...
          'Color', colors(i,:), 'MarkerFaceColor', colors(i,:));
207
208
209
210
211

```

```
212     end
213     [minCycle, maxCycle] = bounds(new_data.CycleNumber);
214
215     for iCycle = minCycle:maxCycle
216         logIndCycle = new_data.CycleNumber == iCycle;
217         radInd = new_data.RadarPosition(logIndCycle);
218         sizes = rescale(new_data.SNR_half_dB(logIndCycle), 5, 40);
219         set(hScatter, 'XData', new_data.Lateral(logIndCycle), ...
220             'YData', new_data.Longitudinal(logIndCycle), ...
221             'ZData', new_data.Heigth(logIndCycle), ...
222             'SizeData', sizes, ...
223             'CData', colors(radInd+1, :), ...
224             'MarkerFaceColor', 'flat');
225         hTitle.String = sprintf('Cycle Number: %d', iCycle);
226         pause(0.05)
227     end
```