

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE**

**Fakulta dopravní**



**DIPLOMOVÁ PRÁCE**

**2016**

**Bc. Lukáš Diblík**

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní



Bc. Lukáš Diblík

**ANALÝZA PŘESNOSTI VYBRANÝCH ZPŮSOBŮ  
URČENÍ ODHADU POLOHY GPS PŘIJÍMAČE**

Diplomová práce

**2016**



**K616.....Ústav dopravních prostředků**

## **ZADÁNÍ DIPLOMOVÉ PRÁCE** (PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

**Bc. Lukáš Diblík**

Kód studijního programu a studijní obor studenta:

**N 3710 – DS – Dopravní systémy a technika**

Název tématu (česky): **Analýza přesnosti vybraných způsobů určení odhadu polohy GPS přijímače**

Název tématu (anglicky): Accuracy Analysis of Selected Methods of GPS Receiver Position Estimation

### **Zásady pro vypracování**

Při zpracování diplomové práce se řiďte osnovou uvedenou v následujících bodech:

- Seznamte se se základními principy fungování systému GPS a základními metodami (Newtonova iterační metoda, Bancroftova algebraická metoda, rozšířený Kalmanův filtr) sloužícími k určení odhadu polohy a času pohybujícího se GPS přijímače.
- S pomocí simulačních dat analyzujte kvalitu poskytovaného odhadu polohy GPS přijímače získaného výše zmíněnými metodami, a to v závislosti na přítomnosti šumu a přítomnosti případných poruch v měřených GNSS datech. Výsledky jednotlivých metod vzájemně porovnejte.
- U výpočtu odhadu polohy přijímače s pomocí rozšířeného Kalmanova filtru analyzujte závislost kvality odhadu polohy přijímače na volbě modelu.
- Navrhněte způsob, jak simulačně získané výsledky ověřit v železničním prostředí.
- Porovnejte simulačně získané výsledky s výsledky získanými z reálných měření z železničního prostředí.

Rozsah grafických prací: dle pokynů vedoucího práce

Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)

Seznam odborné literatury: Misra, P., Enge, P.; Global Positioning System – Signals, Measurements and Performance; 2nd Ed.; USA; 2006.  
Hrdina, Z., Pánek, P., Vejražka, F.; Radiové určování polohy (Družicový systém GPS); Vydavatelství ČVUT; Praha; 1995.  
Šimandl, M.; Identifikace systémů a filtrace; Skriptum ZČU; Plzeň; 2001.

Vedoucí diplomové práce:

**Ing. Milan Sliacky**  
**Ing. Karel Veselý, Ph.D.**

Datum zadání diplomové práce: **29. června 2015**

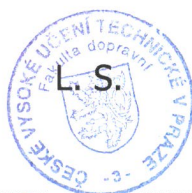
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **30. listopadu 2016**

- a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia  
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia



doc. Ing. Petr Bouchner, Ph.D.  
vedoucí  
Ústavu dopravních prostředků



prof. Dr. Ing. Miroslav Svítek, dr. h. c.  
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.

24.11.2016



Bc. Lukáš Diblík  
jméno a podpis studenta

V Praze dne .....2. června 2016

# Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi poskytli podklady pro vypracování této diplomové práce. Děkuji Ing. Milanu Sliackemu a zejména pak Ing. Karlu Veselému, Ph.D. za jeho odborné vedení, vstřícnost, pečlivost a rady, které mi poskytoval po celou dobu psaní této práce. Rovněž děkuji společnosti AŽD Praha s.r.o. za poskytnutá data a demonstrační jízdu s měřicím železničním vozem. V neposlední řadě je mou milou povinností poděkovat svým rodičům, blízkým a přítelkyni za morální a materiální podporu, které se mi dostávalo po celou dobu studia.

# Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na ČVUT v Praze Fakultě dopravní.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30. listopadu 2016



.....

podpis

**Název práce:** Analýza přesnosti vybraných způsobů určení odhadu polohy GPS přijímače

**Autor:** Bc. Lukáš Diblík

**Obor:** Dopravní systémy a technika

**Druh práce:** Diplomová práce

**Vedoucí práce:** Ing. Karel Veselý, Ph.D

Samostatný vývojový pracovník

Závod Technika, úsek Výzkum a vývoj, VP03

AŽD Praha s.r.o.

**Ing. Milan Sliacky**

Ústav dopravní telematiky K620

Fakulta dopravní

České vysoké učení technické v Praze

## **Abstrakt**

Tématem této diplomové práce je analýza matematických metod, které slouží ke stanovení odhadu polohy a času pohybujícího se GPS přijímače. Práce je zaměřena především na komparaci těchto algoritmů z hlediska přesnosti poskytovaného odhadu polohy s využitím simulčních dat a rovněž údajů získaných z praktických měření. Cílem je také posouzení možné implementace těchto algoritmů do GPS přijímačů v železniční dopravě.

## **Klíčová slova**

Bancroftova metoda, GPS, Kalmanův filtr, MATLAB, Newtonova metoda, odhad polohy, přijímač, signál, železniční doprava

**Title:** Accuracy Analysis of Selected Methods of GPS Receiver Position Estimation

**Author:** Bc. Lukáš Diblík

**Branch:** Transportation Systems and Technology

**Document type:** Master's thesis

**Supervisor:** Ing. Karel Veselý, Ph.D.

Senior Development Engineer

Research & Development

AŽD Praha s.r.o.

**Ing. Milan Sliacky**

Department of Transportation Telematics K620

Faculty of Transportation Sciences

Czech Technical University in Prague

## **Abstract**

The subject of this master's thesis is the analysis of mathematical methods which are applied to GPS receiver position and time estimation. Primarily, it is focused on the comparison of these methods from the point of view of the localization accuracy by means of simulated and real data. One of the objectives is the evaluation of possible implementation of these algorithms in rail transport.

## **Keywords**

Bancroft's Method, GPS, Kalman Filter, MATLAB, Newton's Method, Position Estimation, Railway Transport, Receiver, Signal



# Obsah

Obsah .....	7
Seznam zkratek.....	9
<b>Úvod .....</b>	<b>12</b>
<b>1 Princip družicové navigace .....</b>	<b>13</b>
1.1 SOUŘADNICOVÝ SYSTÉM .....	14
1.2 ČAS .....	15
1.3 RÁDIOVÉ URČENÍ POLOHY .....	16
1.3.1 Družice GPS .....	18
1.3.2 Navigační signály .....	19
1.3.3 Modulace signálů.....	25
1.3.4 Zpracování signálů .....	26
<b>2 Určení odhadu polohy .....</b>	<b>30</b>
2.1 CHYBY MĚŘENÍ.....	30
2.2 MATEMATICKÉ ALGORITMY PRO VÝPOČET ODHADU POLOHY .....	31
2.2.1 Bancroftova algebraická metoda .....	32
2.2.2 Newtonova iterační metoda .....	34
2.2.3 Rozšířený Kalmanův filtr .....	37
2.3 MATICE VAH .....	41
2.4 FAKTOR DOP.....	42
<b>3 Analýza metod určení odhadu polohy .....</b>	<b>44</b>
3.1 PŘÍPRAVA PRO SIMULAČNÍ ANALÝZU .....	46
3.1.1 Skript_Algoritmus.m .....	49
3.1.2 Funkce_PDOP.m .....	51
3.1.3 Funkce_Bancroft.m .....	52
3.1.4 Funkce_Newton.m.....	52
3.1.5 Funkce_KalmanR2.m a Funkce_KalmanR3.m .....	52
3.1.6 Funkce_Elipsoid.m .....	53
3.2 PRŮBĚH A VYHODNOCENÍ SIMULACE .....	53
<b>4 Ověření metod určení odhadu polohy .....</b>	<b>67</b>
4.1 LOKALIZACE VOZIDEL NA ŽELEZNIČNÍCH TRATÍCH.....	67
4.1.1 Zabezpečovací zařízení v železniční dopravě.....	69
4.1.2 Využití GNSS pro lokalizaci drážních vozidel.....	71

---

4.2	NÁVRH POSTUPU REÁLNÉHO MĚŘENÍ.....	73
4.3	PROVEDENÍ REÁLNÉHO MĚŘENÍ .....	77
4.4	VYHODNOCENÍ VÝSTUPŮ REÁLNÉHO MĚŘENÍ .....	79
	<b>Závěr .....</b>	<b>82</b>
	Seznam obrázků.....	83
	Seznam tabulek.....	85
	Seznam použité literatury .....	86
	Seznam příloh.....	91

**Příloha A: Skripty vytvořené v programu MATLAB**

**Příloha B: Výstupy provedených simulací**

# Seznam zkratek

<b>AŽD</b>	Automatizace železniční dopravy
<b>BOC</b>	Binary Offset Carrier
<b>BPSK</b>	Binary Phase Shift Keying
<b>BTS</b>	Base Transceiver Station
<b>C/A (kód)</b>	Coarse Acquisition
<b>CD</b>	Compact Disc
<b>CDMA</b>	Code Division Multiple Access
<b>ČR</b>	Česká republika
<b>ČSN</b>	Česká technická norma
<b>DMI</b>	Driver Machine Interface
<b>DOP</b>	Dilution of Precision
<b>DSSS</b>	Direct Sequence Spread Spectrum
<b>ECEF</b>	Earth Centered Earth Fixed
<b>EKF</b>	Extended Kalman Filter
<b>ERTMS</b>	European Rail Traffic Management System
<b>ETCS</b>	European Train Control System
<b>GDOP</b>	Geometric Dilution of Precision
<b>GLONASS</b>	Globalnaja Navigacionnaja Sputnikovaja Sistěma
<b>GNSS</b>	Global Navigation Satellite System
<b>GPRS</b>	General Packet Radio Service
<b>GPS</b>	Global Positioning System
<b>GPST</b>	Global Positioning System Time
<b>GSM-R</b>	Global System for Mobile Communications - Railway
<b>HDOP</b>	Horizontal Dilution of Precision

<b>HOW</b>	Hand-Over Word
<b>KF</b>	Kalman Filter
<b>KM</b>	Kovarianční matice
<b>M (kód)</b>	Military
<b>MATLAB</b>	Matrix Laboratory
<b>NaN</b>	Not a Number
<b>NMEA</b>	National Marine Electronics Association
<b>P (kód)</b>	Precision
<b>PDOP</b>	Positional Dilution of Precision
<b>PRN</b>	Pseudo Random Noise
<b>QPSK</b>	Quadratic Phase Shift Keying
<b>RBS</b>	Radioblok - Station
<b>RBV</b>	Radioblok - Vehicle
<b>RFID</b>	Radio Frequency Identification
<b>RINEX</b>	Receiver Independent Exchange
<b>SIL</b>	Safety Integrity Level
<b>STD</b>	Standard Deviation
<b>SW</b>	Software
<b>TAI</b>	International Atomic Time
<b>TDOP</b>	Time Dilution of Precision
<b>TEN-T</b>	Trans-European Transport Network
<b>TLM</b>	Telemetry Word
<b>TOW</b>	Time Of Week
<b>TSI</b>	Technické specifikace interoperability
<b>USA</b>	United States of America
<b>UT</b>	Universal Time

<b>UTC</b>	Coordinated Universal Time
<b>VDOP</b>	Vertical Dilution of Precision
<b>VHF</b>	Very High Frequency
<b>WGS 84</b>	World Geodetic System 1984

# Úvod

Jedním ze základních pojmů, jenž se týká družicové navigace, je přesnost polohy uživatele. Existuje mnoho faktorů, které tuto přesnost mohou ovlivnit. Nicméně samotný výpočet polohy je prováděn na základě různých matematických algoritmů, jejichž použití může být za určitých okolností výhodné, nebo naopak nevhodné.

V železniční dopravě je znalost relativně přesné polohy velmi důležité, a to zejména v bezpečnostních aplikacích, kdy rozhoduje přesnost o hodnotě několika jednotek metrů. Nabízí se tedy otázka, jaké matematické algoritmy jsou pro železniční dopravu nejvíce vhodné.

Cílem této práce, iniciované společností AŽD Praha, s.r.o, je tedy analýza těchto algoritmů z hlediska přesnosti a zhodnocení jejich reakce na přítomnost poruch v měřených datech.

Vzhledem ke koncepci je tato práce rozdělena na část teoretickou a část praktickou. V první části se uplatnila především metoda deskriptivní. Nejprve je popsán princip činnosti družicové navigace systému GPS, a to formou od generování signálů na družici až po příjem a zpracování signálů v přijímači na straně uživatele. Poté jsou zevrubně analyzovány jednotlivé matematické algoritmy pro stanovení odhadu polohy GPS přijímače. V další části je rozebrána možná implementace družicové navigace v železniční dopravě spolu s uvedenými úspěšnými příklady ze zahraničí, ale i z České republiky.

Praktická část souvisela s tvorbou matematických algoritmů v programu MATLAB, dále pak se simulacemi a následným zhodnocení daných metod z hlediska přesnosti a reakce na injektované poruchy. Simulace byly provedeny na datech vygenerovaných i naměřených.

# 1 Princip družicové navigace

Již na počátku civilizace si lidé uvědomovali nezbytnost stanovení své polohy v prostoru a nalezení cesty k požadovanému cíli. Význam určení správné trasy výrazně vzrostl v dobách námořních plaveb, kdy bylo potřeba nejen určit pozici plavidla a porovnat ji s předpokládanou polohou, ale také na základě toho rozhodnout o dalším směru plavby. Na pevnině toto obvykle nečinilo značné potíže díky dostatečnému počtu orientačních bodů, ovšem na moři bylo možné se orientovat pouze pomocí nebeských těles. K tomuto účelu byly využívány různé vynálezy, jako například astroláb či sextant. Úmyslem tedy bylo cílevědomě vést plavidlo po předem definované trase. Odtud také vznikl pojem navigace (lat. vést loď). [1, 2]

Díky rozvoji vědy a techniky, zejména v období druhé světové války, se uplatnily pro lokalizaci a navigaci objektů rádiové signály. Vznikaly radionavigační systémy, jejichž základem byla síť pozemních vysílačů (např. systém Loran či Omega). Když v roce 1957 Sovětský svaz vypustil první umělou družici Sputnik, byly jeho signály z vesmíru monitorovány týmem amerických vědců. Ti zjistili, že jej lze relativně přesně lokalizovat na oběžné dráze za předpokladu znalosti jejich přesné polohy na Zemi, a to na základě Dopplerova efektu frekvence vysílaného a přijatého signálu. Tím zároveň vyvstala myšlenka o inverzi tohoto případu. Tedy možnost na základě známé polohy družice na oběžné dráze lokalizovat přijímač na Zemi. [2]

V polovině 60. let 20. století byl uveden do činnosti první družicový navigační systém Transit, který pracoval právě na základě měření Dopplerova posunu frekvence. Nicméně systém disponoval šesti družicemi obíhající Zemi na polární dráze, což umožnilo stanovit polohu uživatele přibližně jednou za hodinu. Na počátku 70. let téhož století začaly Spojené státy americké budovat dokonalejší globální navigační satelitní systém s názvem GPS. [2]

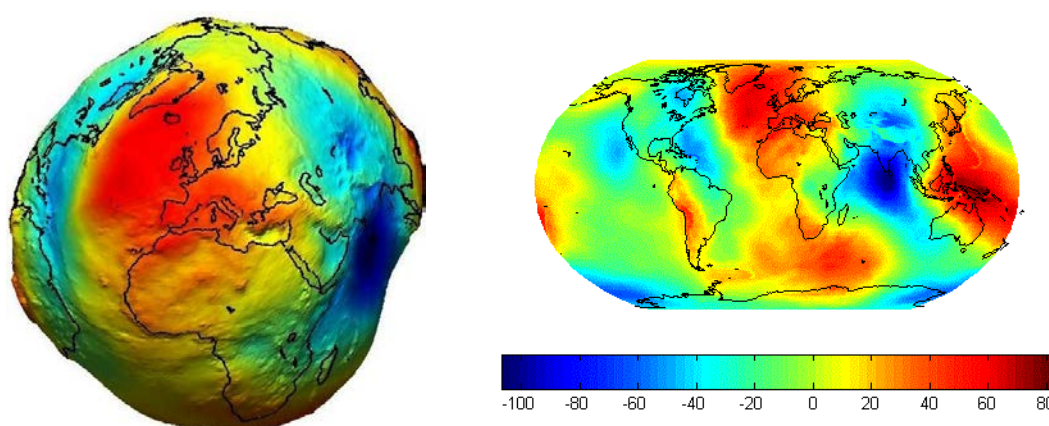
Družicové systémy jsou pro úlohu lokalizace a navigace výhodné oproti pozemním systémům především globálním pokrytím Země pomocí relativně malého počtu vysílačů. Principem pro stanovení polohy je tedy zpracování signálů vysílaných družicemi na oběžných drahách.

Tato kapitola se zabývá základním principem fungování rádiového určení polohy pomocí družic. Jednotlivé podkapitoly jsou záměrně napsány tak, aby na sebe logicky navazovaly. Čili od tvorby navigačních signálů a dat na družici až po konečné zpracování těchto signálů v přijímači. Druhá kapitola, zabývající se výpočtem polohy uživatele, plynule navazuje na tuto část. Nejdříve je ale nezbytné zmínit se obecně o referenčních soustavách a času.

## 1.1 Souřadnicový systém

Pro stanovení polohy kdekoli na Zemi je nezbytné nejprve definovat souřadnicový systém, ke kterému budou vztaženy další výpočty. Pro uživatele, který se nachází na Zemi, je výhodné pracovat s geocentrickým souřadnicovým systémem (angl. ECEF). Avšak družice obíhají naši planetu nezávisle na jejím rotačním pohybu. Tudíž z daného hlediska je výhodnější aplikovat referenční soustavu, jejíž střed je zároveň středem sluneční soustavy. Nicméně pro možnost lokalizace uživatele musí přijímač znát polohu družic na oběžných drahách vztaženou ke stejnému (geocentrickému) souřadnicovému systému. Mezi oběma referenčními soustavami tedy existuje přesná matematická transformace, prováděná přímo v uživatelském zařízení, která mimo jiné zohledňuje precesi a nutaci naší planety. [3, 4]

Povrch planety Země nejněběji vystihuje fyzikální model nazvaný geoid. Geoid je definován jako plocha, která je kolmá v každém bodě k tíhovému zrychlení při střední hladině moří protažené pod kontinenty. Bohužel kvůli nehomogenitě a vlivu vlastní rotace Země je tento model matematicky obtížně definovatelný, tudíž se pro kartografické potřeby nahrazuje rotačním elipsoidem. Toto nahrazení nijak neovlivní určení zeměpisné šířky a délky, ovšem skutečná nadmořská výška je odlišná. Pro získání výšky nad geoidem je tedy nezbytné elipsoidickou výšku následně přepočítat. Pro Českou republiku se difference mezi výškou nad geoidem a elipsoidem pohybuje kolem 45 m. Tvar geoidu je zachycen na níže uvedeném obrázku, ve kterém jsou barevně rozlišeny výškové rozdíly geoidu oproti rotačnímu elipsoidu. Tmavě modrá barva značí rozdíl -105 m, zatímco rudá barva značí diferenci +85 m. [4, 5]

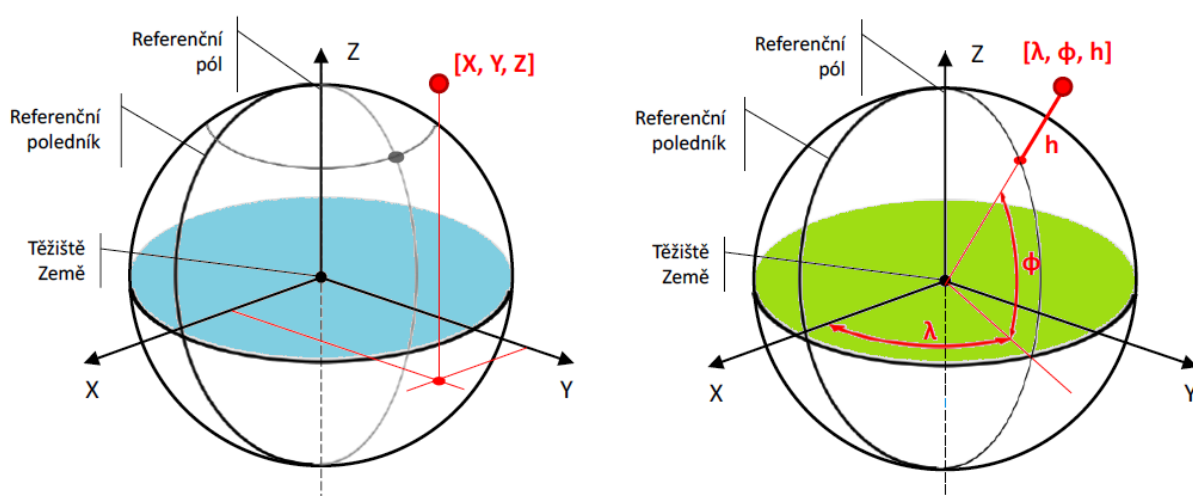


Obr. 1: Fyzikální model povrchu Země - geoid [6, 7]

System GPS používá referenční soustavu označenou WGS 84. Tento světově uznávaný geodetický souřadnicový systém nahrazuje tvar Země specifickým rotačním elipsoidem. Ačkoli



přijímač zobrazuje uživateli polohu v zeměpisných (geodetických) souřadnicích, vypočítává svou polohu v kartézském souřadném systému. Počátek kartézské soustavy je v těžišti Země. V této soustavě je pozice uživatele určena jako vzdálenost od daného počátku ve 3 směrech ( $x, y, z$ ). Kladná osa  $x$  směřuje k průsečíku rovníku a nultého poledníku, kladná osa  $z$  míří k severnímu pólu a kladná osa  $y$  je na obě kolmá, tak aby soustava byla pravotočivá. Určení souřadnic v této referenční soustavě je sice výpočetně podstatně jednodušší, ovšem je následně nutná transformace do zeměpisných sférických souřadnic. U sférických souřadnic je poloha stanovena jako úhel od rovníku, úhel od nultého poledníku a výška nad geoidem. Rozdíl mezi kartézskými a geodetickými souřadnicemi je zobrazen na obrázku č. 2. [5, 8]



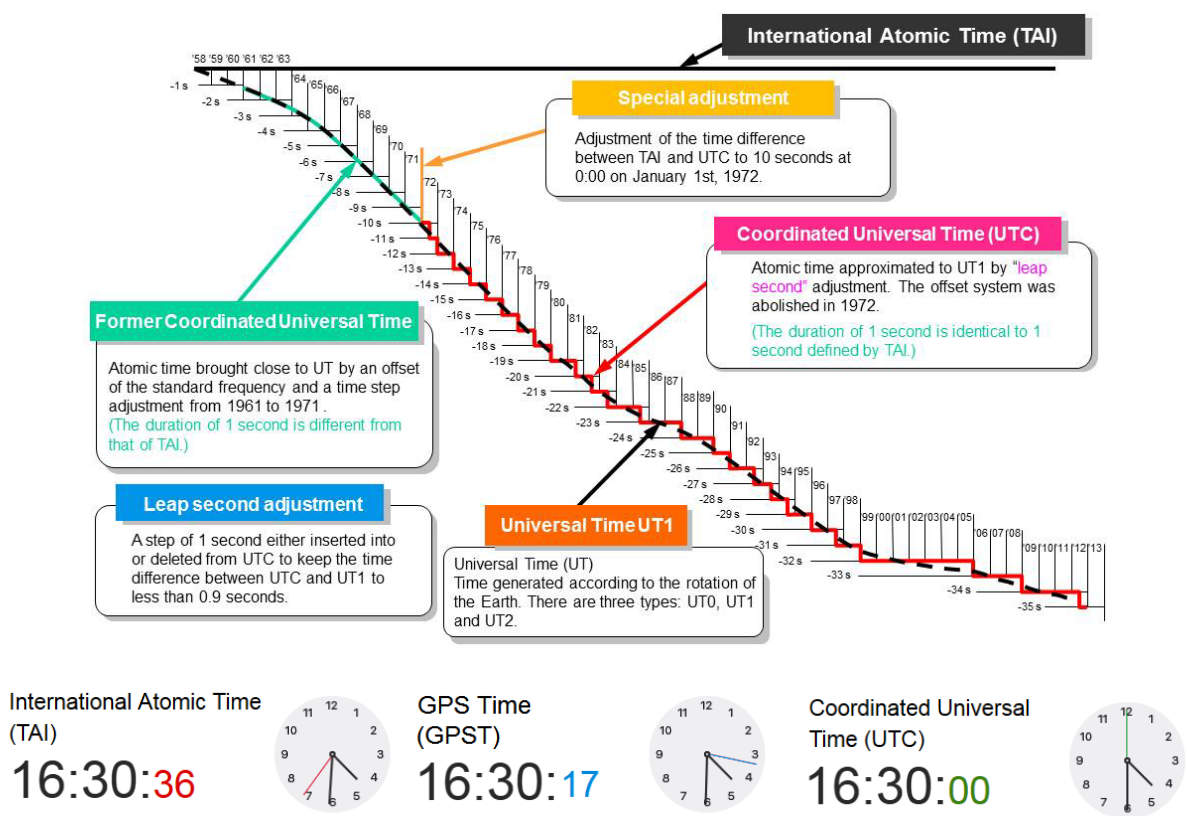
Obr. 2: Kartézské a geodetické souřadnice [2, upraveno]

## 1.2 Čas

Pro korektní fungování celého družicového navigačního systému hraje zásadní roli základní fyzikální veličina, a tou je čas. Pro přesné určení času používá kosmický a řídicí segment GPS atomové hodiny, které využívají rezonanční frekvenci atomů cesia a rubidia. Čas odvozený od atomových hodin bývá označen zkratkou TAI. Díky jejich fyzikální podstatě nejsou hodiny nikterak závislé na rotačním pohybu Země. Avšak rotace naší planety se postupně zpomaluje (zhruba o 0,9 sekund za rok), čili atomový čas není synchronní s astronomickým časem (UT). Astronomický čas je ale odvozený od relativně nepravidelného rotačního pohybu Země, čili je také nepravidelný, a tudíž nepřilíš dokonalý pro přesné stanovení času. Z tohoto důvodu byl zaveden čas UTC, který je odvozován od atomových hodin, ovšem je korigován tak, aby byl v souladu s astronomickým časem. Korekce je prováděna přičtením, nebo odečtením tzv.

přestupné sekundy (angl. leap second) k času UTC tak, aby byl vždy zajištěn rozdíl mezi UT a UTC pod hodnotou 0,9 s. V současné době je čas TAI napřed o 36 sekund oproti UTC. [4]

Čas systému GPS (GPST) je odvozován od času TAI, kdy od roku 1980 již nepoužívá GPST přestupné sekundy. TAI je v současné době o 19 s napřed oproti GPST a tento rozdíl je již konstantní. V přijímačích je tato diference opravována pomocí SW. GPST je tedy atomový čas, jenž se začal určovat 6. ledna 1980 v 0:00:00 času UTC. Čas GPS je uváděn v týdnech od daného data a v sekundách od počátku každého týdne. Nový týden začíná vždy o půlnoci ze soboty na neděli. Družice vysílá směrem k uživateli také údaje, na jejichž základě je umožněn přepočítání GPST na UTC. Na obrázku č. 3 je pro názornost zobrazen vývoj časů TAI, UT a UTC od roku 1958 do roku 2013. [1, 9]



Obr. 3: Čas TAI, GPST a UTC [10, 11, upraveno]

### 1.3 Rádiové určení polohy

Stanovení polohy na Zemi prostřednictvím systémů GNSS funguje na principu vysílání signálů družicemi a jejich následného přijetí a zpracování uživatelskými zařízeními. Vzhledem k tomu, že komunikace probíhá pouze jednosměrně od družice k uživateli, jedná se o pasivní systém. Signálem se v tomto případě rozumí fyzikální veličina, pomocí níž lze přenášet data.

V případě navigačních signálů jde o rádiové vlny, jejichž rychlost v prostoru je přibližně rovna rychlosti světla ve vakuu. Pro výpočet polohy uživatele lze aplikovat několik metod, přičemž v současné době převládá u systémů GNSS metoda dálkoměrná založená na principu trilaterace. [2]

U dálkoměrné metody je základem stanovení doby šíření signálu od družice k uživateli. Družice vyšle signál s časovou značkou, kdy byl odeslán. Čas odeslání je v uživatelském zařízení porovnán s časem přijetí daného signálu. Pokud se tento tranzitní čas vynásobí rychlostí šíření rádiových vln (tedy rychlostí světla), je vypočtena vzdálenost mezi družicí a přijímačem. Pro lokalizaci uživatele na Zemi je ale nutné znát polohu družic na oběžných drahách. V prostoru lze tedy na základě měření k jedné družici s jistotou určit, že se uživatel nachází kdesi na kulové ploše, v jejíž středu je družice a poloměrem je vypočtená vzdálenost. Pokud bude provedeno měření k další družici, bude se uživatel nacházet na průniku dvou kulových ploch, tedy kdesi na kružnici. Teprve až měření ke třetí družici pomůže stanovit polohu uživatele. Třetí kulová plocha totiž protíná danou kružnici ve dvou bodech. Jeden se obvykle nachází ve vesmíru a druhý je hledaná poloha uživatele. Jinými slovy pro vypočtení třech neznámých  $X, Y, Z$  je zapotřebí tři rovnic, a tudíž postačí měření ke třem družicím. Toto ale platí pouze v ideálním případě, kdy hodiny přijímače a družice jsou dokonale synchronizované. [3, 12]

V praxi je ale mezi hodinami družice (tedy časem GPST) a přijímače jistá časová diference, která po vynásobení rychlostí šíření rádiových vln zanášá do určení pozice chybu ve vzdálenosti. Z tohoto důvodu je tedy místo skutečné vzdálenosti vypočtena tzv. pseudovzdálenost (zdánlivá vzdálenost). Tento časový nesoulad představuje v rovnicích čtvrtou neznámou. Pro výpočet čtyř neznámých jsou nutné již čtyři rovnice. To je tedy důvod, proč je nezbytně nutné pro stanovení polohy v prostoru měření alespoň ke čtyřem družicím. Výpočet polohy probíhá v kartézských souřadnicích a je definován níže uvedenými zjednodušenými rovnicemi

$$P_1 = \sqrt{(x_r - x_{s1})^2 + (y_r - y_{s1})^2 + (z_r - z_{s1})^2} + ct \quad (1.1)$$

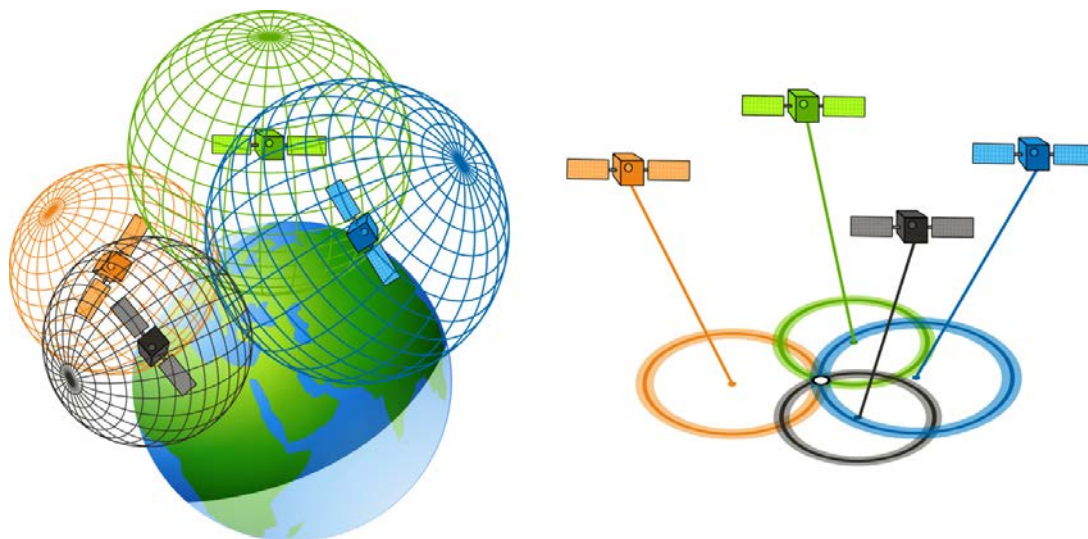
$$P_2 = \sqrt{(x_r - x_{s2})^2 + (y_r - y_{s2})^2 + (z_r - z_{s2})^2} + ct \quad (1.2)$$

$$P_3 = \sqrt{(x_r - x_{s3})^2 + (y_r - y_{s3})^2 + (z_r - z_{s3})^2} + ct \quad (1.3)$$

$$P_4 = \sqrt{(x_r - x_{s4})^2 + (y_r - y_{s4})^2 + (z_r - z_{s4})^2} + ct, \quad (1.4)$$

kde na levé straně rovnic jsou hodnoty  $P_i$  pseudovzdálenosti. Pod odmocninou představují proměnné  $x_r, y_r, z_r$  dočasně neznámé kartézské souřadnice přijímače. Souřadnice  $x_{si}, y_{si}, z_{si}$  reprezentují polohu družic na oběžných drahách opět v kartézském souřadném systému.

Parametry oběžných drah družic jsou vysílány samotnými družicemi k uživateli. Příjímač z těchto parametrů následně vyhodnotí danou polohu družice. Proměnná  $t$  charakterizuje výše zmíněný časový posun hodin přijímače od družice. Konstanta  $c$  označuje rychlost světla. Pro zobrazení zeměpisných sférických souřadnic je v přijímači provedena transformace vypočtených kartézských souřadnic. Ačkoli je výše uvedeno, že čas družice je shodný se systémovým časem GPST, jde pouze o idealizaci v této práci. Ve skutečnosti jsou i tyto časy mírně odlišné, nicméně řídicí segment zasílá uživateli mimo jiné parametry pro jejich korekci. [1]



Obr. 4: Princip určení polohy pomocí čtyř družic [13, 14]

### 1.3.1 Družice GPS

V současné době disponuje systém GPS 31 družicemi, které obíhají planetu Zemi na střední oběžné dráze ve výšce 20 200 km. Družice jsou rozmístěné v šesti orbitálních drahách s inklinací  $55^\circ$ , což zaručí, že z každého místa na Zemi jsou viditelné alespoň čtyři družice. U dálkoměrné metody hraje velmi významnou roli stanovení času, tudíž hlavní komponentou všech navigačních družic jsou velmi přesné hodiny. Satelity GPS mívají dva páry rubidiových a cesiových atomových hodin. [15]

V předchozí části je vysvětleno, že pro stanovení polohy uživatele je nezbytnou podmínkou nejprve určit polohu družic na oběžných drahách, což je v praxi značně obtížné. Satelity se pohybují rychlostí kolem 3,8 km/s a jejich pohyb lze popsat pomocí Keplerových zákonů. Každá družice vysílá směrem k uživateli mimo jiné svou polohu, tedy přesněji jde o parametry orbitální dráhy (tzv. efemeridy), pomocí níž je v přijímači poloha družice vypočtena. Veškeré signály, jež jsou družicemi vysílány, jsou odvozeny od základní frekvence s hodnotou 10,23 MHz. Tato hodnota vychází z frekvence družicových atomových hodin. [3, 8]

Vzhledem k tomu, že satelity obíhají Zemi relativně vysoko a značnou rychlostí, podléhá jejich pohyb relativistickým efektům. Speciální teorie relativity říká, že čas plyne pro pozorovatele na rychle pohybujícím se objektu pomaleji, než pro pozorovatele v klidu. Díky tomu se družicové hodiny zpožďují o 7 milisekund za den. Naopak podle obecné teorie relativity plynou hodiny umístěné dál od hmotného objektu (např. planety Země) rychleji oproti hodinám bližším k danému objektu. Díky tomuto jevu hodiny družic předbíhají o 45 milisekund za den. Celkově tedy hodiny na družicích předbíhají o 38 milisekund za den, což by odpovídalo chybě ve vzdálenosti 10 km každý den. Kvůli kompenzaci relativistických jevů jsou neustále řídicím segmentem prováděny korekce hodin a zároveň je základní frekvence snížena na hodnotu 10,229 999 995 450 MHz, která se ale pro pozorovatele na Zemi jeví jako 10,23 MHz. Nutno dodat, že přesnost hodin je pro lokalizaci uživatele nejdůležitější, poněvadž například chyba 1 mikrosekundy v měřeném čase způsobí chybu 300 m ve vzdálenosti. [16, 17]

### 1.3.2 Navigační signály

Jak již bylo dříve uvedeno, výpočet polohy uživatele u družicové navigace je prováděn na základě zpracování signálů vysílaných družicemi. Tyto signály, jež jsou kombinací nosné vlny, dálkoměrného kódu a navigační zprávy, jsou vysílány kontinuálně. [18]

#### Nosné vlny

Frekvence nosných vln byla zvolena z pásma L, které se pohybuje mezi 1 - 2 GHz. Vlnová délka signálů s hodnotou frekvence v daném rozmezí odpovídá zhruba 20 cm. U signálů s těmito vlnovými délkami bývá relativně nízká interference s ionosférou, což je pro přesné určení polohy podstatné. Systém GPS pracuje v současnosti s pěti kmitočty, jež jsou odvozeny od základní frekvence atomových hodin, tedy od hodnoty 10,23 MHz. Pro navigační účely slouží frekvence L1 (1 575,42 MHz) a frekvence L2 (1 227,60 MHz). Již šest let vysílají družice také na frekvenci L5 (1 176,45 MHz), která slouží výlučně pro účely bezpečnosti v letecké dopravě. Tato hodnota totiž spadá do kmitočtového pásma, které je vyhrazené pro leteckou radionavigaci. To znamená, že tato frekvence nesmí být rušena jinými signály, ovšem přijímat ji může kdokoli. Frekvence L3 a L4 slouží pouze pro řídicí segment [4, 19]

#### Dálkoměrné kódy

Součástí signálu je vyjma nosné vlny také dálkoměrný kód. Jak již jeho název napovídá, díky tomuto kódu je umožněno dálkoměrné měření. Systém GPS již od počátku pracuje se dvěma základními kódy nesoucí označení C/A (angl. Coarse Acquisition) a P (angl. Precision).

Vzhledem k tomu, že systém GPS je vojenský projekt, pracují družice navíc s kódem M (angl. Military), který je ale využíván pouze pro armádní účely a nebude mu tudíž v této práci věnována pozornost. Pomocí dálkoměrných kódů je v přijímači určen časový rozdíl, na jehož základě je proveden výpočet zdánlivé vzdálenosti. Podrobněji se této problematice věnuje kapitola týkající se zpracování signálů. [19, 20]

Aby mohl přijímač určit pseudovzdálenost k jednotlivým družicím, musí být nejdříve schopen rozlišit signály od daných satelitů. Toho je docíleno také dálkoměrnými kódy, které jsou pro každou družici jedinečné. Všechny družice tedy vysílají signály na stejném kmitočtu, avšak každá používá svůj unikátní dálkoměrný kód. Jde tedy o kódový multiplex (CDMA). [21, 22]

Dálkoměrné kódy u systému GPS nesou obecné označení PRN (angl. Pseudo Random Noise), poněvadž představují na první pohled náhodnou posloupnost jedniček a nul a podobají se tak náhodnému šumu. Nahodilost je ale ve skutečnosti pouze zdánlivá, poněvadž PRN kódy vznikají dle exaktních matematických vztahů. Svými číselnými charakteristikami se tyto série jedniček a nul pouze podobají náhodné posloupnosti. V kapitole zabývající se zpracováním signálů bude vysvětleno, proč tyto PRN kódy nemohou mít stochastický charakter. [16, 22]

### **Kód C/A**

Dálkoměrný kód C/A je definován jako pseudonáhodná posloupnost o délce 1 023 jedniček a nul. Tento kód moduluje frekvenci L1 a je přístupný pro běžného uživatele bez poplatků. Každá družice vysílá svůj jedinečný C/A kód, kterému je přiděleno PRN číslo. Díky tomuto číslu jsou v přijímači jednotlivé družice identifikovány a zároveň od sebe odlišeny. Kmitočet kódu má hodnotu 1,023 MHz, tudíž jeho rychlost odpovídá hodnotě 1,023 Mbit/s. Z toho plyne, že se C/A kód opakuje každou jednu milisekundu. [4, 8]

Pro kódový multiplex je nejvýhodnější použít posloupnost ze skupiny tzv. Goldových kódů. Sekvence C/A kódů vznikají binárním součtem dvou stejně dlouhých polynomů G1 a G2. Tyto tzv. dílčí kódy jsou tvořeny pomocí posuvného registru s deseti buňkami. Každá buňka obsahuje pouze jeden bit (tedy hodnotu 1, nebo 0). Nutno dodat, že dálkoměrné kódy nepřenášejí žádná data, tudíž v tomto případě nejsou hodnoty nazývány bity, nýbrž chipy. Kvůli velikosti registru, jenž je tvořen deseti buňkami, je omezena délka kódu na hodnotu 1 023 bitů (chipů), což vychází z relace  $2^{10} - 1$ . [4, 23]

Polynom dílčího kódu  $G1$  je definován ve tvaru

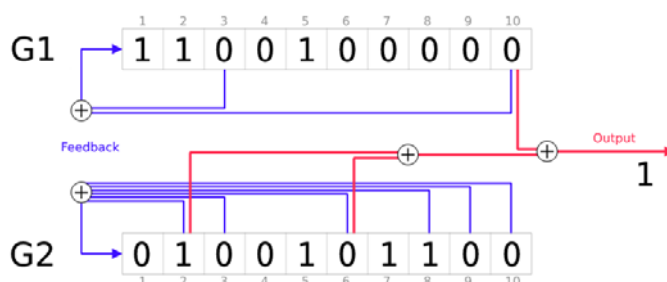
$$G1 = 1 + X^3 + X^{10} \quad (1.5)$$

a pro polynom druhého dílčího kódu  $G2$  platí

$$G2 = 1 + X^2 + X^3 + X^6 + X^8 + X^9 + X^{10}, \quad (1.6)$$

kde  $X$  značí příslušný chip (hodnotu 1, nebo 0) a horní index pouze označuje pozici v registru, na které se daný chip nachází. Schematické znázornění tvorby C/A kódu je znázorněno na níže uvedeném obrázku. V posuvném registru polynomu  $G1$  probíhá operace binárního součtu (modulo-2) chipů, které se nachází na 3. a 10. pozici. Pokud je součet sudé číslo, přepíše se na hodnotu 0, naopak v případě lichého výsledku se zapíše hodnota 1. Tato hodnota (0 nebo 1) se zapíše na první pozici posuvného registru, a tím se všechny hodnoty chipů v registru posunují o jednu buňku doprava. Hodnota na desáté pozici je výstupem z polynomu  $G1$ . V dalším kroku je opět proveden součet chipů na 3. a 10. pozici. Tento iterační proces se opakuje 1 023x, poté jsou všechny buňky v registru nastaveny na hodnotu 1 (resetovány). Po resetování probíhá celý proces znovu. [3, 8, 12]

Dílčí kód  $G2$  je generován obdobným postupem jako kód  $G1$ , ovšem zde probíhá součet chipů, jež se nachází na 2., 3., 6., 8., 9. a 10. pozici. Jejich součet je opět zapsán na první pozici registru, čímž se ostatní hodnoty opět posouvají o jednu buňku doprava. Nicméně u polynomu  $G2$  není již výstupem 10. hodnota, nýbrž kombinace dvou buněk, specifických pro každou družici. Jednotlivým satelitům je přiřazena hodnota PRN, díky které jsou odlišeny od ostatních. Například družice s identifikátorem PRN 1 odpovídá kombinaci buněk 2 a 6 v posuvném registru dílčího kódu  $G2$ . Pro družici PRN 5 jde o kombinaci buněk 1 a 9. Binární součet buněk 2 a 6 v polynomu  $G2$  spolu se součtem výstupní hodnoty polynomu  $G1$  vytvoří první chip dálkoměrného kódu pro družici s označením PRN 1. Iterační proces polynomu  $G2$  se opět opakuje 1 023x a poté je resetován. Tímto opakovaným postupem, který trvá pouhou jednu milisekundu, vzniká kompletní dálkoměrný C/A kód. [3, 8, 12]



Obr. 5: Princip generování kódu C/A [24, upraveno]

## **Kód P**

Na rozdíl od kódu C/A, moduluje dálkoměrný kód P oba nosné kmitočty L1 a L2 a je vysílán s frekvencí 10,23 MHz. Příjem na dvou kmitočtech značně redukuje ionosférickou refrakci, čímž se stává měření přesnější. Ovšem tento kód je přístupný pouze autorizovaným uživatelům (např. pro geodetické účely). Oproti kódu C/A je výrazně delší, poněvadž jeho celková délka činí zhruba 266 dní. Družice jsou odlišeny tímto kódem tak, že každá vysílá pouze jeho část. Délka kódu je tedy rozdělena na 38 částí o délce 7 dní a každá tato část je vysílána jinou družicí. Kód C/A je opakován každou jednu milisekundu, kdežto kód P se opakuje po sedmi dnech. Měření je přesnější také z důvodu použití delšího a rychlejšího kódu (jeho frekvence je 10x rychlejší než u kódu C/A). Tento kód může být i zašifrovaný pomocí kódu W, poté nese označení Y. Dešifrování probíhá v přijímači opět pomocí kódu W, se kterým ale pracují pouze vojenské přijímače. [5, 8]

## **Ostatní signály**

Nutno dodat, že v rámci modernizace jsou v systému GPS přidávány další signály. Již deset let je zpřístupněn civilní signál s označením L2C, jenž je přenášen na frekvenci L2 a je určen výhradně pro komerční účely. Nový signál L1C (na frekvenci L1) bude přístupný během tohoto roku a bude interoperabilní s ostatními GNSS systémy. [4, 8]

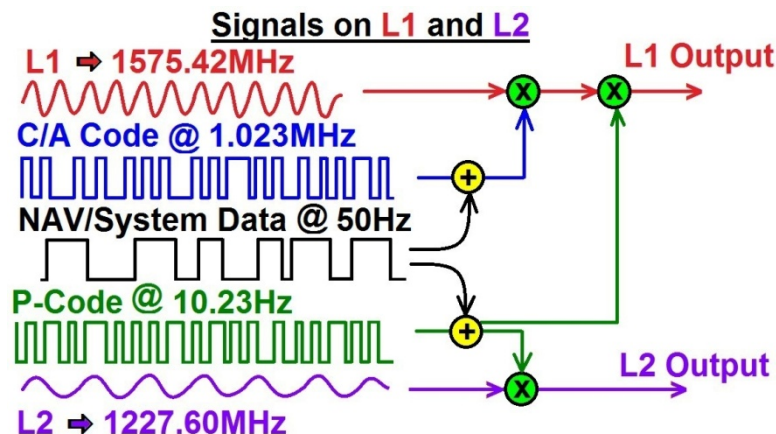
## **Navigační zpráva**

Dálkoměrné kódy vysílané družicemi tedy slouží k odlišení jednotlivých satelitů a k následnému určení zdánlivé vzdálenosti přijímače od dané družice. Nicméně k tomu, aby byl přijímač schopen určit svou polohu, musí znát nejprve pozici družice, k níž je pseudovzdálenost měřena. Proto je, kromě dálkoměrných kódů, součástí signálu také navigační zpráva, ve které jsou obsažena potřebná data. [18, 21]

Kmitočet navigační zprávy je rovněž odvozován od atomových hodin družice a má hodnotu 50 Hz, čili rychlost odpovídá 50 bit/s. Tato zpráva je binárně sečtena (modulo-2) s dálkoměrným kódem, čímž vznikne signál s rozprostřeným spektrem, který je následně přidán k nosné vlně. Vzhledem k této rychlosti je zpráva přenášena spolu s dálkoměrným kódem pomocí techniky DSSS, což značí, že jeden bit zprávy je přenesen pomocí několika chipů kódu (např. pomocí 20 460 chipů kódu C/A). Signál je tedy rozprostřen do větší části rádiového spektra, čímž je méně citlivý vůči rušení, což zvyšuje spolehlivost přenosu. Takovýto signál se poté uživateli jeví jako šum a přenášená data je velmi obtížné získat bez znalosti původní sekvence kódu C/A. Výsledný signál je poté vyslán směrem k uživateli. Generování signálů zachycuje

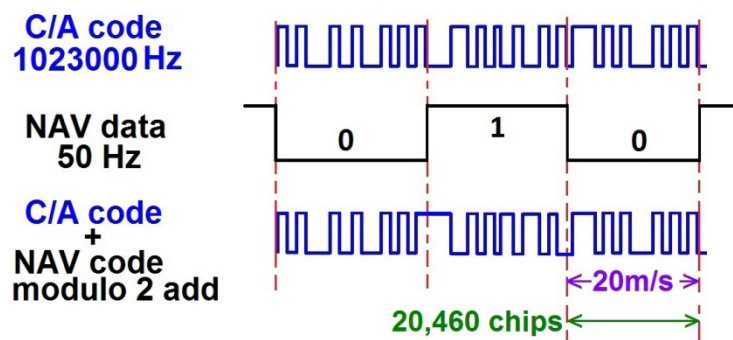


přehledně obrázek č. 6, kde znaménko (+) značí binární součet a znaménko (x) reprezentuje směšovač. Výstupem směšovače je signál o frekvenci, která je dána součtem či rozdílem kmitočtů vstupních signálů. [18, 21]



Obr. 6: Princip generování signálů GPS [25, upraveno]

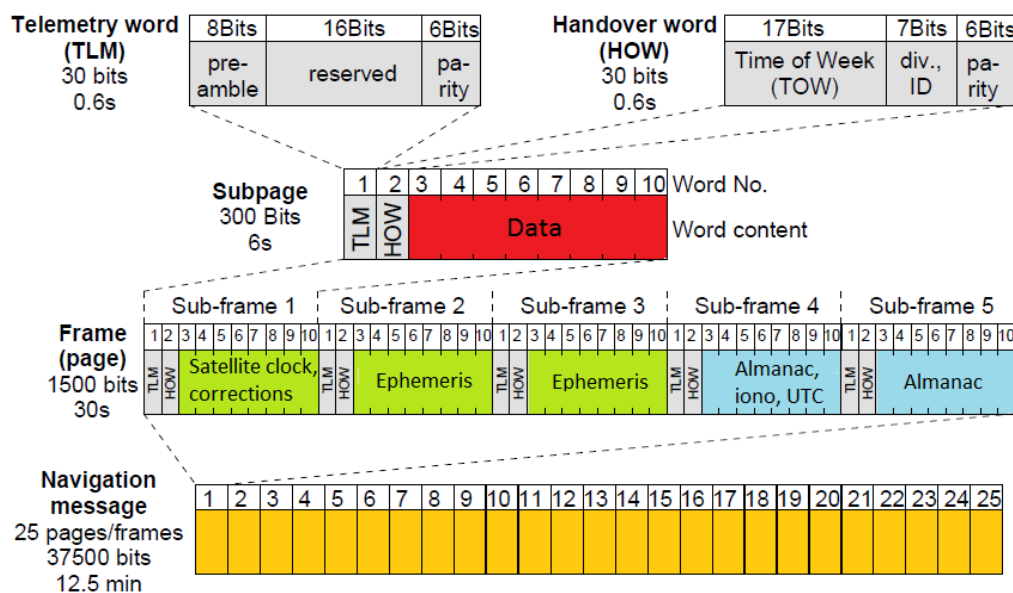
Na dalším obrázku je zobrazena binární modulace navigační zprávy a dálkoměrného kódu C/A. Zpráva je modulována na dálkoměrný kód na základě jednoduchého principu. Má-li bit v navigační zprávě nulovou hodnotu, zůstává C/A kód nezměněn. Avšak je-li hodnota bitu ve zprávě rovna jedné, dojde k převrácení kódu.



Obr. 7: Princip modulace navigační zprávy a kódu [25, upraveno]

Navigační data jsou rozdělena do 25 rámců a skládají se celkem ze 37 500 bitů. Kvůli relativně nízké přenosové rychlosti trvá vysílání jedné kompletní zprávy 12,5 minuty. Každý rámeček, jehož velikost je 1 500 bitů, je rozdělen do 5 podrámčů po 300 bitech. Podrámček obsahuje dále 10 slov o délce 30 bitů, přičemž pro přenos dat je využito pouze 24 bitů. Posledních šest bitů slouží k zabezpečení přenosu dat proti chybám (paritní bity). K tomuto zabezpečení používá systém GPS Hammingův kód, což je významný kód používaný v oblasti telekomunikací.

Je totiž schopen detekovat dvojnásobnou chybu, případně opravit jednonásobnou chybu. Struktura celé navigační zprávy je zobrazena níže. [8, 23]



Obr. 8: Struktura navigační zprávy GPS [26, upraveno]

V navigační zprávě jsou potřebné údaje odděleny v jednotlivých podrámcích. První tři podrámcce jsou pro každou družici jedinečné a jsou vysílány každých 30 sekund (tedy po každém odeslání celého jednoho rámcce). Zatímco čtvrtý a pátý je společný všem družicím a jeho update probíhá po 12,5 minutách. [8]

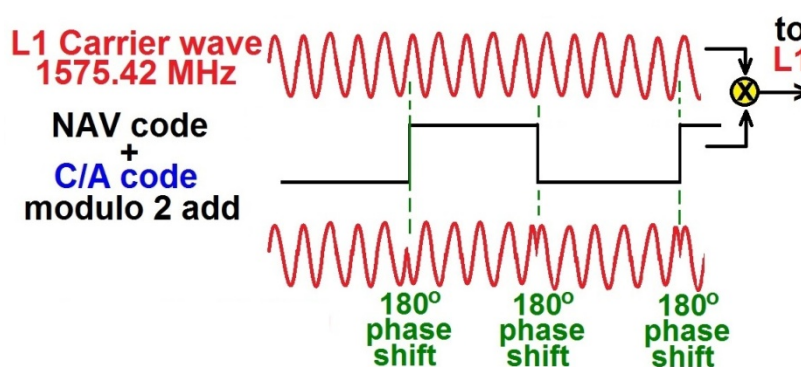
V prvním podrámcce jsou uvedeny parametry pro korekci družicových hodin a stav družice. Druhý a třetí podává informaci o poloze družice. Nicméně nejde přímo o konkrétní polohu, nýbrž o velmi přesné parametry oběžné dráhy (efemeridy), ze kterých je posléze v přijímači poloha vypočtena. Obsah čtvrtého a pátého podrámcce se týká údajů o ionosférickém modelu, čase UTC a zejména jsou zde méně přesné parametry oběžných drah všech družic systému GPS (almanach). Koeficienty ionosférického modelu jsou v přijímači využity pro snížení ionosférické refrakce, což je důležité zejména v případě jednofrekvenčního měření. Díky znalosti aktuálního almanachu je výrazně snížena doba nutná k prvnímu určení polohy po zapnutí přijímače. Efemeridy jsou platné po dobu čtyř hodin, kdežto parametry ionosférického modelu včetně almanachu mívají platnost i několik dní. [8, 12]

Na začátku každého podrámcce jsou první dvě slova stejná. První TLM (angl. Telemetry Word) přenáší informaci určenou pro řídicí segment, zatímco druhé slovo HOW (angl. Hand-Over Word) nese informaci o pořadí následujícího podrámcce od začátku GPS týdne (TOW). Zbýlých osm slov se týká dat uvedených v předchozích odstavcích. [12, 27]

### 1.3.3 Modulace signálů

Navigační zpráva spolu s dálkoměrným kódem tvoří diskrétní data, konkrétně posloupnost jedniček a nul. Ovšem aby bylo možné takovýto signál přenést rádiovou cestou, je nutné jej konvertovat na spojitý. To je také nezbytné pro zajištění správného průchodu signálu atmosférou, a tím dosažení přesnějšího měření. Z daného důvodu jsou diskrétní data modulována na nosnou vlnu o určité frekvenci. Modulace modifikuje parametry spojitě vlny (frekvenci, amplitudu nebo fázi) na základě vstupního diskrétního znaku. Obecně lze tedy říci, že díky modulaci je diskrétní signál převeden na spojitý. [23]

System GPS pracuje se dvěma typy modulace. Pro navigační zprávu s dálkoměrným kódem C/A nebo P je použito tzv. binární fázové klíčování (BPSK), které spočívá ve změně počáteční fáze nosné vlny o hodnotu  $180^\circ$  při každé změně bitu diskrétního signálu (kód se zprávou). Díky tomu, že jsou bity (1 a 0) od sebe snadno rozlišeny převrácením nosné frekvence o  $180^\circ$ , je BPSK velmi odolná proti chybám. Tato modulace je také efektivní tím, že při minimálním výkonu je možné přenést signál na relativně velkou vzdálenost. Princip je tedy obdobný jako u modulace kódu a zprávy a je zobrazen na obrázku č. 9. [3, 23]



Obr. 9: Modulace BPSK [25, upraveno]

Vysílané signály lze poté v závislosti na čase zapsat ve tvaru

$$L1(t) = a_1 P(t) D(t) \cos 2\pi(f_1 t) + a_2 C(t) D(t) \sin 2\pi(f_1 t) \quad (1.7)$$

$$L2(t) = a_3 P(t) D(t) \cos 2\pi(f_2 t), \quad (1.8)$$

kde  $L1$  a  $L2$  jsou výsledné signály vysílané družicemi,  $P$  a  $C$  reprezentují dálkoměrné kódy,  $D$  značí navigační zprávu,  $a_i$  představuje amplitudu a  $f_i$  frekvenci signálů. Vzhledem k tomu, že  $L1$  je modulován oběma kódy, používá se pro jejich oddělení vzájemně ortogonální posunutí (o  $90^\circ$ ). Celkově tedy signál  $L1$  představuje čtyřstavovou modulaci (QPSK). [9, 23]

Za zmínku také stojí modulace označená zkratkou BOC. Takto je modulován například dálkoměrný kód M a tuto modulaci bude využívat i evropský navigační systém Galileo. Pro BOC modulaci je autokorelační funkce užší než pro BPSK, a tím se měření stává přesnějším. Na druhou stranu má ale BOC více maxim autokorelační funkce, což může způsobit jistou nejednoznačnost při zpracování signálu v přijímači. [8, 28]

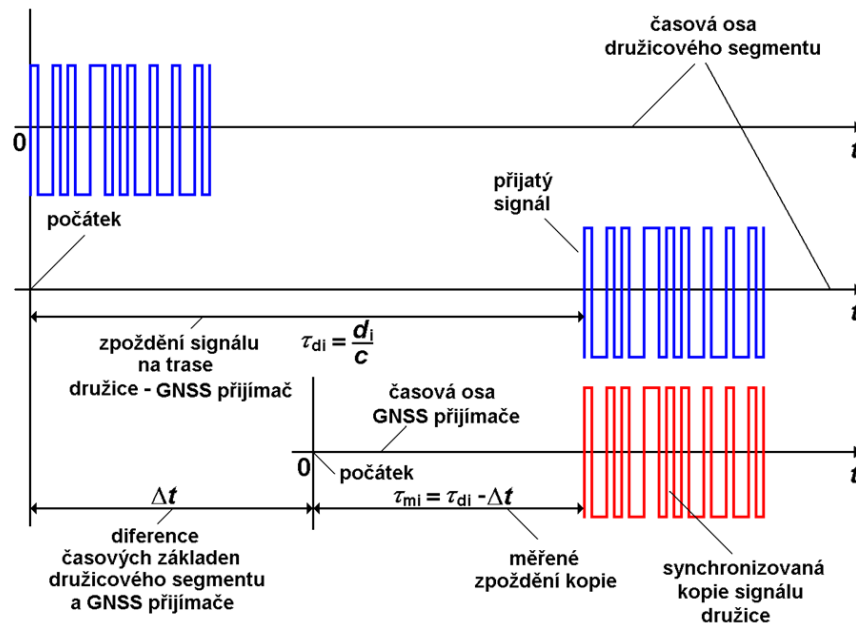
### 1.3.4 Zpracování signálů

Hlavní účel přijímače je pochopitelně stanovení polohy pomocí zpracovaných signálů vysílaných navigačními družicemi. Ovšem aby bylo možné polohu vypočítat, musí být přijímač schopen zvládnout několik klíčových úkolů. V první řadě identifikovat družici, ke které bude následně měřit pseudovzdálenost, poté určit tranzitní čas (doba přenosu signálu), ze kterého vypočte zdánlivou vzdálenost a následně vytěžit údaje ze signálu příslušné družice. Díky matematickým algoritmům, z již známých dat, je vypočtena poloha uživatele. To je ale předmětem následující kapitoly.

#### Určení zdánlivé vzdálenosti

Tranzitní čas je určen na základě zpoždění přijatého signálu. Jednoduše lze říci, že dojde k porovnání času vysílání a času přijetí. Časový rozdíl poté odpovídá tranzitnímu času signálu. Avšak ve skutečnosti je princip složitější. Přijímač totiž generuje kopii dálkoměrného kódu, která je poté časově synchronizována s kódem přijatým. Každá družice vysílá jedinečný dálkoměrný kód (CDMA), tudíž je nezbytné, aby uživatelské zařízení znalo kódy všech družic v systému GPS. Pokud je v přijímači vygenerována kopie přijatého signálu, dojde k posunutí této kopie na pomyslné časové ose o takový počet bitů, aby došlo k jejímu ztotožnění se signálem přijatým. Tímto nastává korelace. Vynásobením potřebného počtu bitů pro synchronizaci signálů s dobou trvání jednoho bitu je získána doba šíření signálu. To je také důvod, proč PRN kódy nemohou mít náhodný charakter. Přijímač by totiž nedokázal vygenerovat stejný kód jako daná družice. Obrázek č. 10 zachycuje princip synchronizace kopie signálu s přijatým signálem. [9]

Následně stačí tranzitní čas signálu vynásobit rychlostí šíření rádiových vln, díky čemuž je stanovena vzdálenost přijímače od družice. Synchronizace obou signálů je provedena v přijímači, tudíž se vztahuje k počátku časové základny uživatele. Právě kvůli odlišným časovým osám přijímače a systému GPST (pro zjednodušení uvažován jako čas družice), vystupuje v rovnici čtvrtá neznámá, která je následně zjištěna díky měření ke čtyřem a více družicím. [9]



Obr. 10: Synchronizace přijatého signálu s jeho kopií [8]

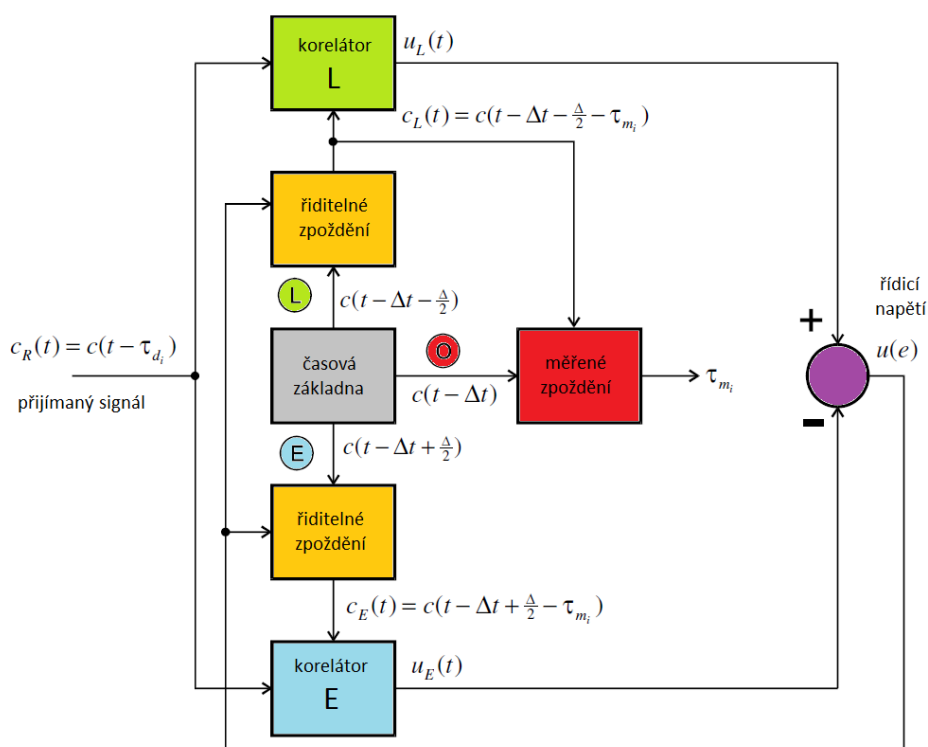
### Diskriminátor zpoždění

Dálkoměrné kódy se neustále opakují s jistou periodou, což u kratších kódů (např. C/A) může způsobit určitou nejednoznačnost při měření tranzitního času. Přijímač totiž nemusí rozeznat, o kolik period byl signál na trase zpožděn. Tato nejistota je eliminována užitím diskriminátoru zpoždění, který obsahuje dva korelátory a jeho architektura je na obrázku č. 11. [19, 21]

Pomocí diskriminátoru zpoždění dochází k posuvu kopie signálu. Tento posuv je řízen napětím  $u(e)$ , jehož velikost a znaménko odpovídá velikosti a smyslu potřebného posuvu. Signál přijatý od družice, čili zpožděný o hodnotu tranzitního času  $\tau_{di}$ , je přiváděn na oba korelátory. Zároveň do obou těchto korelátorů přichází kopie daného signálu (generovaná v přijímači), zpožděná o hodnotu  $\Delta t$  oproti signálu přijatému kvůli rozdílné časové základně družice a přijímače. Ovšem do jednoho z těchto korelátorů přichází kopie signálu s určitým zpožděním o hodnotu  $\Delta/2$ , zatímco na vstupu druhého korelátoru generovaná kopie o hodnotu  $\Delta/2$  předbíhá. V tomto případě značí písmeno  $\Delta$  délku bitu vysílaného signálu. [19, 21]

Následně vytvoří korelátor součet součinů hodnot signálu přijatého s jeho kopií. Tento součet součinů se nazývá autokorelační funkce, jejíž hodnota se odvíjí od vzájemného posuvu těchto signálů. V případě nulového posuvu, kdy jsou tedy signály totožné, je hodnota korelační funkce maximální. Zároveň hodnota autokorelační funkce dvou různých signálů má velmi malou hodnotu, takže nemůže dojít k zasynchronizování dvou odlišných kódů. Je-li přijatý signál blíže k zasynchronizování se signálem v korelátoru L, bude napětí  $u_L$  větší než  $u_E$ .

To způsobí, že výsledné řídicí napětí  $u(e)$  bude kladné a bude působit na smyčku takovým způsobem, že se generovaná kopie bude více zpožďovat, dokud nedojde ke ztotožnění signálů. Naopak, je-li přijatý signál blíže ke kopii vstupující do korelátoru E, bude napětí  $u_E$  větší, a tedy výsledné řídicí napětí  $u(e)$  bude záporné. Tímto dojde ke zvětšení předbíhání kopie opět do doby, než se oba signály nezasynchronizují. Pokud je kopie signálu totožná se signálem vysílaným družicí, bude napětí na obou korelátorech stejné, tudíž řídicí napětí bude nulové. V tomto případě přijímač získá měřené zpoždění kopie  $\tau_{mi}$ , ze kterého je stanovena pseudovzdálenost. [19, 21]

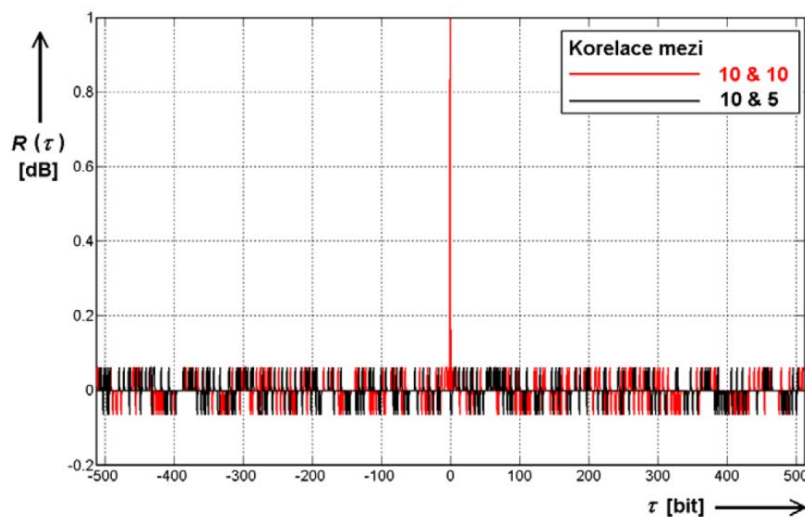


Obr. 11: Diskriminátor zpoždění [19, upraveno]

### Autokorelační funkce

Díky naměření maximální hodnoty autokorelační funkce je zaručeno, že vzájemné posunutí obou signálů je nulové a je tudíž možné stanovit zdánlivou vzdálenost. Aby takovéto určení bylo jednoznačné, je nezbytné, aby autokorelační funkce měla pouze jedno ostré maximum a zároveň aby hodnota této funkce byla nízká pro dva odlišné signály. Tyto podmínky právě splňují Goldovy kódy spolu s modulací BPSK. Obrázek č. 12 demonstruje autokorelační funkci pro kopii kódu družice s identifikátorem PRN 10 a přijatého kódu z družice PRN 10 (červený průběh). Je-li vzájemný posun těchto signálů nulový, je naměřeno jedno ostré maximum. Pro porovnání je na obrázku také zobrazen vztah, reprezentován černou barvou, opět

pro kopii kódu družice PRN 10, ale v korelaci se signálem vysílaným družicí PRN 5. Lze tedy spatřit, že pro dva různé signály je hodnota autokorelační funkce relativně velmi nízká. Díky tomu, že vzájemné korelace odlišných kódů mají nízké hodnoty, mohou být všechny tyto kódy přenášeny na stejné frekvenci, aniž by se ovlivňovaly (princip CDMA). [8]



Obr. 12: Autokorelační funkce [8]

### Získání dat z navigační zprávy

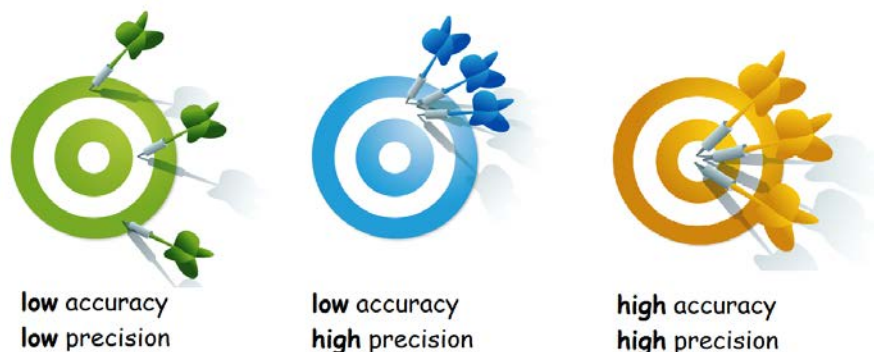
Po kódovém měření musí dojít k demodulování signálu, aby se přijímači zpřístupnila navigační zpráva. Signál s navigační zprávou a dálkoměrným kódem C/A, matematicky  $C(t)D(t)$ , je vynásoben s příslušným kódem (tedy C/A). Díky tomu, že  $C^2(t)$  je rovno jedné, dojde k odfiltrování dálkoměrného kódu a zůstane pouze navigační zpráva. Přijímač poté extrahuje ze zprávy důležitá data, která využije pro výpočet polohy družice na oběžné dráze či pro korekci hodin družice apod. [9]

## 2 Určení odhadu polohy

Dalo by se říci, že demodulace signálu je relativně bezproblémová operace, tudíž vypočtené souřadnice přijímače by měly korespondovat se skutečnými hodnotami. Ovšem součástí přijatého signálu jsou také nežádoucí rušivé složky (např. šum atmosférický nebo šum vnitřních obvodů uživatelského zařízení). Kvůli tomu a zejména kvůli chybám měření nelze přesně rekonstruovat přijatý signál a v přijímači probíhá pouze odhadování diskretních znaků s určitou mírou pravděpodobnosti, a tedy odhadování polohy uživatele. Problematice výpočtu polohy z naměřených údajů a získaných dat se věnuje tato kapitola. [23]

### 2.1 Chyby měření

Chyby tedy pochopitelně značně ovlivňují přesnost. Vzhledem k tomu, že přesnost je zásadní veličina nejen pro tuto práci, ale obecně pro lokalizaci, je vhodné objasnit rozdíl v anglické a české terminologii. Anglický výraz pro *přesnost* je *precision*, nicméně v anglické literatuře se pro přesnost v souvislosti s lokalizací používá spíše termín *accuracy*, jenž odpovídá českému výrazu *správnost*. Rozdíl ve významu těchto termínů je patrný z následujícího obrázku.



Obr. 13: Správnost a přesnost [29, upraveno]

Níže uvedená rovnice reprezentuje chyby při měření vzdálenosti mezi družicí a přijímačem, respektive vztah mezi skutečnou vzdáleností a vzdáleností naměřenou

$$P = R + \Delta_{ion} + \Delta_{trop} + \Delta_{clk_s} + \Delta_{clk_r} + \Delta_{rel} + \Delta_{noi} + \Delta_r, \quad (2.1)$$

přičemž  $P$  představuje naměřenou pseudovzdálenost a  $R$  skutečnou vzdálenost. Výčet chyb v rovnici směrem zleva je vliv ionosféry, vliv troposféry, chyba hodin družice, chyba hodin přijímače, vliv relativistických efektů, vliv šumu měření a další vlivy (mnohacestné šíření signálu, apod.). Z těchto faktorů má velký vliv na nepřesnost ionosféra, poněvadž obsahuje



relativně vysoký počet iontů a elektronů. Rádiové signály, vysílané družicemi, mají charakter elektromagnetického záření, tudíž jsou při průchodu atmosférou přítomnými elektricky nabitými částicemi značně ovlivňovány. Signál se v této vrstvě láme, čili doba šíření je delší a přijímač poté naměří větší pseudovzdálenost. Obecně lze říci, že chyba způsobená vlivem ionosférické refrakce má podíl kolem 50 % z celkové chyby určení polohy. Nutno dodat, že šum měření nebývá považovaný za chybu, ale spíše za přirozenou složku. [4, 30]

V následujících oddílech této práce se předpokládá, že výše uvedené chyby jsou kompenzovány. Bude uvažován pouze šum měření, díky čemuž půjde o stochastický systém.

## 2.2 Matematické algoritmy pro výpočet odhadu polohy

Všechny metody odhadu polohy přijímače, jež jsou popsány v této práci, využívají naměřené pseudovzdálenosti mezi přijímačem a družicí a údaje o poloze družic získaných z navigační zprávy. Při přijetí faktu, že GPS je systém stochastický, je možné měřenou pseudovzdálenost vyjádřit následující rovnicí. Ta neobsahuje výše popsané chyby, protože se pro zjednodušení předpokládá, že je již provedena jejich korekce

$$P_i = \sqrt{(x_r - x_{si})^2 + (y_r - y_{si})^2 + (z_r - z_{si})^2} + b + \xi, \quad (2.2)$$

kde proměnné  $x_r, y_r, z_r$  jsou souřadnice přijímače (angl. receiver) a hodnoty  $x_{si}, y_{si}, z_{si}$  odpovídají souřadnicím dané družice (angl. satellite). Parametr  $b$  označuje tzv. bias a v rovnici představuje čtvrtou neznámou. Tento bias, neboli systematická chyba, vzniká důsledkem již dříve zmíněné difference časové základny přijímače od systémového času GPST. Náhodná proměnná  $\xi$  představuje šum měření (chyba odhadu). Pro šum se předpokládá normální rozdělení s nulovou střední hodnotou a rozptylem daným číslem  $\sigma^2$ . [16, 31]

Existuje mnoho metod pro výpočet polohy, ovšem s ohledem na zadání je tato práce zaměřena na tři nejčastěji využívané. Nejdříve je popsána neiterativní metoda, poté iterativní metoda a třetím analyzovaným algoritmem je hojně využívaný Kalmanův filtr. U některých metod je nutné před samotným výpočtem a priori odhadnout neznámé souřadnice přijímače. Jak má být tento prvotní odhad stanoven, není vždy zcela zřejmé, přitom na přesnosti odhadu závisí výpočetní doba skutečných souřadnic. Pro většinu pozemních, lodních a leteckých aplikací je prvotní odhad jednoduše určen na základě vizualizace oblasti, ve které se daná aplikace nachází (např. USA, Evropa apod.). [32]

### 2.2.1 Bancroftova algebraická metoda

Tento neiterativní algoritmus byl poprvé publikován fyzikem jménem Stephen Bancroft v roce 1985. Vychází především ze skutečnosti, že rovnice pro jednotlivá měření pseudovzdáleností jsou si podobné. V jeho článku, týkajícího se této metody, je uvedeno, že stanovení polohy uživatele pomocí následujícího algoritmu je v případě špatného geometrického uspořádání družic přesnější než u iterativní metody, což by mělo být simulačně ověřeno v této práci. Odlišnost oproti níže popsaným metodám je ale především v tom, že Bancroftův algoritmus nevyužívá Taylorův rozvoj a není vyžadován a priori odhad polohy přijímače. [33]

Je nutné definovat vektor souřadnic družice a přijímače. Oba vektory jsou sloupcové, nicméně v této práci budou sloupcové vektory převážně zapsány jako transponované řádkové vektory. Souřadnice družice jsou již před vlastním výpočtem polohy uživatele vypočteny z parametrů obsažených v navigační zprávě a jejich vektor je definován jako

$$\mathbf{r}_{si} = (x_{si}, y_{si}, z_{si})^T \quad (2.3)$$

a vektor souřadnic přijímače je následující

$$\mathbf{r}_r = (x_r, y_r, z_r)^T. \quad (2.4)$$

Pro výpočet Bancroftovy metody je nutné rovnici (2.2) algebraicky upravit na tvar bez odmocniny a následně přeuspořádat proměnné. Šum je již součástí naměřené pseudovzdálenosti.

$$x_r^2 - 2x_r x_{si} + x_{si}^2 + y_r^2 - 2y_r y_{si} + y_{si}^2 + z_r^2 - 2z_r z_{si} + z_{si}^2 = P_i^2 - 2P_i b + b^2 \quad (2.5)$$

$$(x_{si}^2 + y_{si}^2 + z_{si}^2 - P_i^2) - 2(x_r x_{si} + y_r y_{si} + z_r z_{si} - P_i b) + (x_r^2 + y_r^2 + z_r^2 - b^2) = 0, \quad (2.6)$$

kde první závorka představuje pozici družice včetně příslušné naměřené pseudovzdálenosti, čili jde o známé parametry. Zatímco ve třetí závorce jsou uvedeny souřadnice přijímače včetně biasu, tedy neznámé parametry. Pro další formální zjednodušení zápisu bude výhodné aplikovat tzv. Lorentzovu metriku Minkowského časoprostoru, která je uplatňována v teorii relativity a pro čtyřrozměrný prostor má následující obecný tvar

$$\langle \mathbf{a}, \mathbf{b} \rangle = a_1 b_1 + a_2 b_2 + a_3 b_3 - a_4 b_4, \quad (2.7)$$

kde  $a_i$  a  $b_i$  jsou jednotlivé složky vektorů  $\mathbf{a}$  a  $\mathbf{b}$ . Vektory souřadnic družice a přijímače jsou doplněny o parametr příslušné pseudovzdálenosti, příp. biasu. Poté lze vektory zapsat takto

$$\mathbf{s}_{si} = (x_{si}, y_{si}, z_{si}, P_i)^T \quad (2.8)$$

$$\mathbf{s}_r = (x_r, y_r, z_r, b)^T, \quad (2.9)$$

kde  $\mathbf{s}_r$  je právě hledaný vektor neznámých parametrů, a tudíž výstupem této metody. Výše uvedené vektory souřadnic přijímače a družice lze přepsat ve formátu vyplývajícího z definice Lorentzovy metriky

$$\langle \mathbf{s}_{si}, \mathbf{s}_{si} \rangle = x_{si}^2 + y_{si}^2 + z_{si}^2 - P_i^2 \quad (2.10)$$

$$\langle \mathbf{s}_r, \mathbf{s}_r \rangle = x_r^2 + y_r^2 + z_r^2 - b^2, \quad (2.11)$$

kde tyto rozepsané výrazy jsou shodné s tvary uvedenými v rovnici (2.6), tudíž lze dosadit Lorentzovu metriku do vztahu (2.6), který bude následně vydělen číslem 2

$$\frac{1}{2} \langle \mathbf{s}_{si}, \mathbf{s}_{si} \rangle - \langle \mathbf{s}_r, \mathbf{s}_{si} \rangle + \frac{1}{2} \langle \mathbf{s}_r, \mathbf{s}_r \rangle = 0. \quad (2.12)$$

Pro další část algoritmu je nezbytné provést substituci členů předchozí rovnice

$$\frac{1}{2} \langle \mathbf{s}_{si}, \mathbf{s}_{si} \rangle = a_i \quad (2.13)$$

$$\frac{1}{2} \langle \mathbf{s}_r, \mathbf{s}_r \rangle = L. \quad (2.14)$$

Dosud uvedený zápis platí pro měření k jedné družici s označením  $i$ . Dále ale již bude bráno v úvahu měření k více družicím, tudíž je nutné přejít do maticového vyjádření daných relací. Zde je právě využita podobnost vztahů pro měření pseudovzdálenosti. Ze substituovaného  $a_i$  v rovnici (2.13) kvůli více družicím vznikne vektor  $\mathbf{a}$ . Avšak parametr  $L$  se týká pouze jednoho přijímače, čili tato metrika zůstane skalárem. Nicméně aby bylo možné dále pokračovat v algoritmu, je nezbytné, aby veškeré výrazy byly maticové. Tudíž je tento skalár  $L$  vynásoben sloupcovým jednotkovým vektorem  $\mathbf{e}$ , jenž má stejný počet řádků jako vektor  $\mathbf{a}$ . Místo jedné rovnice (2.12) bude tolik rovnic, kolik je provedeno měření. Všechny rovnice (2.12) lze poté díky substituci maticově vyjádřit jako

$$\mathbf{a} - \mathbf{B}\mathbf{s}_r + L\mathbf{e} = 0, \quad (2.15)$$

kde matice  $\mathbf{B}$  obsahuje všechny složky vektoru  $\mathbf{s}_{si}$  a způsobuje obecně přechod od jedné rovnice k  $n$  rovnicím (měření pseudovzdálenosti k  $n$  družicím)

$$\mathbf{B} = \begin{pmatrix} x_{s1} & x_{s2} & x_{s3} & \dots & x_{sn} \\ y_{s1} & y_{s2} & y_{s3} & \dots & y_{sn} \\ z_{s1} & z_{s2} & z_{s3} & \dots & z_{sn} \\ P_1 & P_2 & P_3 & \dots & P_n \end{pmatrix}^T. \quad (2.16)$$

Pro osamostatnění  $\mathbf{s}_r$ , tedy neznámého vektoru se souřadnicemi polohy přijímače a časového rozdílu hodin, je nutná úprava tzv. pseudoinverzní maticí, poněvadž v případě měření k více než čtyřem družicím není matice  $\mathbf{B}$  čtvercová.

$$\mathbf{B}^T \mathbf{B} \mathbf{s}_r = \mathbf{B}^T (\mathbf{a} + L \mathbf{e}) \quad (2.17)$$

$$\mathbf{s}_r = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T (\mathbf{a} + L \mathbf{e}), \quad (2.18)$$

kde výraz  $(\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T$  může být označen jako  $\mathbf{B}^+$ . Bohužel takto zatím nelze hodnoty souřadnic přijímače vypočítat, poněvadž neznámý vektor  $\mathbf{s}_r$  je obsažen rovněž ve skaláru  $L$ . Ovšem tento problém lze algebraicky relativně jednoduše vyřešit dosazením rovnice (2.18) do vztahu (2.14), čímž vznikne výraz

$$L = \frac{1}{2} \langle \mathbf{B}^+ (\mathbf{a} + L \mathbf{e}), \mathbf{B}^+ (\mathbf{a} + L \mathbf{e}) \rangle, \quad (2.19)$$

což po úpravě dává tvar

$$L^2 \langle \mathbf{B}^+ \mathbf{e}, \mathbf{B}^+ \mathbf{e} \rangle + 2L (\langle \mathbf{B}^+ \mathbf{e}, \mathbf{B}^+ \mathbf{a} \rangle - 1) + \langle \mathbf{B}^+ \mathbf{a}, \mathbf{B}^+ \mathbf{a} \rangle = 0. \quad (2.20)$$

Vztah (2.20) představuje standardní kvadratickou rovnici s kořeny  $L_1$  a  $L_2$ . Po výpočtu je nutné tyto kořeny dosadit do rovnice (2.18). Výstupem jsou tedy dvě řešení vektoru  $\mathbf{s}_r$ , přičemž jedno z nich lze vyloučit, poněvadž jde o souřadnice bodu ležícího mimo Zemi, tedy kdesi ve vesmíru. Toto ověření potenciálně správného vektoru je provedeno dosazením obou výsledných vektorů  $\mathbf{s}_r$  do vztahu (2.2). Verifikací vypočtených souřadnic a biasu daná metoda končí. Algoritmus popsany v této práci byl sepsán a formálně upraven na základě analýzy z více podkladů. [31, 32, 33, 34, 35, 36, 37]

## 2.2.2 Newtonova iterační metoda

Cílem tohoto algoritmu je minimalizovat kritérium nelineárních nejmenších čtverců. Newtonova metoda v tomto algoritmu řeší právě problém nelinearity. Jde tedy o metodu nejmenších čtverců řešenou pomocí Newtonovy iterační metody. V publikacích lze tento algoritmus nalézt také pod názvem Newton-Raphson metoda. Cílem je minimalizace sumy mocnin residuí, která představují rozdíl mezi odhadovanou a skutečnou hodnotou. Čím menší bude rozdíl, tím více odhad odpovídá skutečnosti. Opět tato metoda využívá relaci (2.2). [38]

$$P_i = \sqrt{(x_r - x_{si})^2 + (y_r - y_{si})^2 + (z_r - z_{si})^2} + ct + \xi, \quad (2.21)$$

kde proměnné mají stejný význam jako u předchozí metody. Podmínkou Newtonovy iterační metody je, před začátkem vlastního algoritmu, a priori stanovení odhadu neznámých parametrů (souřadnice), neboli predikce. Tento odhad vstupuje do algoritmu a je následně iteračním procesem opravován vypočítanými korekcemi až do té doby, dokud není korekce nižší než předem definovaná hodnota. Rovněž u této metody je šum dále uvažován jako součást naměřené pseudovzdálenosti. [39]

Vektor skutečných souřadnic a skutečné časové difference je

$$\mathbf{s}_r = (x_r, y_r, z_r, t)^T. \quad (2.22)$$

Vektor predikcí souřadnic a časového rozdílu je následující

$$\mathbf{s}_o = (x_0, y_0, z_0, t_0)^T. \quad (2.23)$$

Vektor korekcí souřadnic a časové difference je zapsán jako

$$\Delta \mathbf{s} = (\Delta x_r, \Delta y_r, \Delta z_r, \Delta t)^T. \quad (2.24)$$

Vzhledem k tomu, že skutečná souřadnice je získána jako aktualizace predikce hodnotou korekce, lze napsat obecný vztah této funkce jako

$$(x_r, y_r, z_r, t) = f(x_0 + \Delta x_r, y_0 + \Delta y_r, z_0 + \Delta z_r, t_0 + \Delta t), \quad (2.25)$$

přičemž tuto funkci lze aproximovat pomocí Taylorova polynomu. Newtonova metoda předpokládá jen lineární aproximaci, tudíž řešení se omezí pouze na první řád polynomu. Rozvoj pak vypadá takto

$$f(x_0, y_0, z_0, t_0) + \frac{\partial f(x_0, y_0, z_0, t_0)}{\partial x_0} \Delta x_r + \frac{\partial f(x_0, y_0, z_0, t_0)}{\partial y_0} \Delta y_r + \frac{\partial f(x_0, y_0, z_0, t_0)}{\partial z_0} \Delta z_r + \frac{\partial f(x_0, y_0, z_0, t_0)}{\partial t_0} \Delta t. \quad (2.26)$$

Po vypočtení výše uvedených partiálních derivací je vztah následující

$$P_i = P_0 + \frac{x_0 - x_{si}}{\sqrt{D}} \Delta x_r + \frac{y_0 - y_{si}}{\sqrt{D}} \Delta y_r + \frac{z_0 - z_{si}}{\sqrt{D}} \Delta z_r + c \Delta t, \quad (2.27)$$

přičemž  $D$  je substituce za výraz

$$D = (x_0 - x_{si})^2 + (y_0 - y_{si})^2 + (z_0 - z_{si})^2. \quad (2.28)$$

Dále je provedena substituce za zlomky v (2.27) a rozdíl  $P_i - P_0$  je nahrazen proměnnou  $l_i$ .

$$l_i = a_{xr}^i \Delta x_r + a_{yr}^i \Delta y_r + a_{zr}^i \Delta z_r + c \Delta t, \quad (2.29)$$

což lze pro měření k  $n$  družicím zapsat také maticově následujícím vztahem

$$\begin{pmatrix} l_1 \\ l_2 \\ l_3 \\ \vdots \\ l_n \end{pmatrix} = \begin{pmatrix} a_{xr}^1 & a_{yr}^1 & a_{zr}^1 & c \\ a_{xr}^2 & a_{yr}^2 & a_{zr}^2 & c \\ a_{xr}^3 & a_{yr}^3 & a_{zr}^3 & c \\ \vdots & \vdots & \vdots & \vdots \\ a_{xr}^n & a_{yr}^n & a_{zr}^n & c \end{pmatrix} * \begin{pmatrix} \Delta x_r \\ \Delta y_r \\ \Delta z_r \\ \Delta t \end{pmatrix}. \quad (2.30)$$

Cílem je, aby rozdíl mezi predikovanou hodnotou  $P_0$  a skutečnou hodnotou  $P_i$  byl minimální. Časový rozdíl hodin družice a přijímače je pro všechna pozorování stejný, je tedy možné a zároveň výpočetně výhodnější dosadit za hodnoty  $c$  číslo 1. Rovnici (2.30) lze zapsat jako

$$\mathbf{L} = \mathbf{A}\Delta\mathbf{s}. \quad (2.31)$$

Zde  $\mathbf{L}$  představuje vektor observací,  $\mathbf{A}$  je matice směrových kosinů a  $\Delta\mathbf{s}$  je vektor neznámých korekcí. Pro osamostatnění  $\Delta\mathbf{s}$  je třeba provést stejnou úpravu pseudoinverzní maticí jako u Bancroftovy algebraické metody, poněvadž pokud bude provedeno měření k více než čtyřem družicím, nebude matice  $\mathbf{A}$  čtvercová.

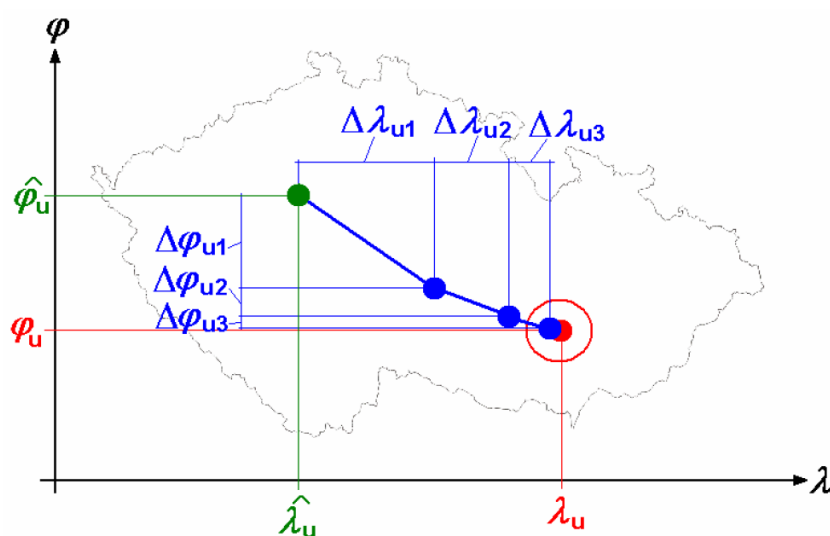
$$\mathbf{A}^T \mathbf{L} = \mathbf{A}^T \mathbf{A} \Delta\mathbf{s} \quad (2.32)$$

$$\Delta\mathbf{s} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{L}. \quad (2.33)$$

Výstupem tohoto algoritmu je tedy vektor korekcí  $\Delta\mathbf{s}$ , jehož složky budou opravovat vstupní a priori odhadnuté parametry souřadnic a časového rozdílu. Viz funkce (2.25).

Newtonův algoritmus je iterační proces. To značí, že výsledné aktualizované hodnoty prvotních odhadů jsou nyní vstupem do další iterace a celý tento výše uvedený proces se opakuje až do té doby, dokud nebude hodnota korekcí dostatečně malá. Iterační proces bývá obvykle značně rychlý a většinou postačuje pouze několik kroků pro dosažení velmi nízké chyby. Nutno dodat, že u této metody je možné odhad charakterizovat pomocí rozptylu.

Níže uvedený obrázek zachycuje odhad, iterace a vypočtenou polohu přijímače ve sférických souřadnicích na území České republiky. A priori odhadnuté souřadnice (zelená barva) reprezentují Prahu, postupnými korekcemi odhadu (modrá barva) je výsledná pozice stanovena jako souřadnice pro město Brno (červená barva). Opět byl tento algoritmus sepsán a formálně upraven na základě analýzy z mnoha podkladů. [35, 38, 39, 40, 41]



Obr. 14: Iterační proces odhadu polohy [8]

### 2.2.3 Rozšířený Kalmanův filtr

Kalmanův filtr (KF) je predikčně-estimační rekurzivní algoritmus, který odhaduje stav dynamického systému. Dynamický systém bývá ale ve skutečnosti ovlivněn šumem stavu a šumem měření. Cílem algoritmu je tedy získání čistého signálu ze zašuměného. KF je složen ze dvou fází, jsou jimi predikce a filtrace (někdy nazývána jako update). Získané výsledky z fáze predikce jsou srovnány s naměřenými vzorky a jejich rozdíl je dále použit ve fázi filtrace ke zlepšení odhadu aktuálního stavu. Níže popsaný KF pracuje s diskretním časem, tedy celá procedura se opakuje každý časový krok (epochu). Výhodou Kalmanova filtru je mimo jiné jeho rychlost. KF předpokládá normální rozložení stavu a výstupu. Takovéto rozložení je plně charakterizováno střední hodnotou a rozptylem. KF tedy pracuje pouze s těmito charakteristikami. Dále je předpokládáno, že v systému je bílý šum. Cílem této práce není odvození Kalmanova filtru, nicméně s ohledem na zadání a na analýzu předchozích dvou metod by se pouhý výčet rovnic mohl jevit jako nedostatečný. [42]

#### Standardní Kalmanův filtr

Hlavním úkolem této metody je získání relativně přesného odhadu skutečného stavu systému. Stav stochastického systému nelze měřit, ovšem lze jej odhadnout. V následujících relacích je dolním indexem  $k$  reprezentována současná epocha, zatímco index  $k-1$  značí předchozí epochu. Stav systému může být popsán touto rovnicí

$$x_k = ax_{k-1} + w_k, \quad (2.34)$$

přičemž  $x$  značí stav,  $a$  představuje parametr a  $w$  je šum stavu (procesu). Z rovnice tedy vyplývá, že současný stav je pochopitelně závislý na předchozím stavu. V tomto případě není uvažováno řízení. Pozorovatel ale získá naměřený stav popsáný rovnicí

$$y_k = hx_k + z_k, \quad (2.35)$$

kde  $y$  představuje pozorovaný (naměřený) stav,  $h$  opět značí parametr a  $z$  znamená šum měření. Rovnice (2.34) a (2.35) reprezentují stavový model systému. Právě kvůli šumu, jehož hodnotu nelze nijak určit, je stav pouze odhadován. Predikované veličiny budou dále značeny stříškou ( $\hat{\phantom{x}}$ ). Update stavu, který vychází jak z predikovaného stavu, tak z naměřených hodnot lze zapsat

$$\hat{x}'_k = \hat{x}_k + K_k(y_k - h\hat{x}_k), \quad (2.36)$$

kde apostrof značí, že jde o updatovaný stav a  $K$  označuje Kalmanův zisk (někdy nese označení Kalmanův gain či zesílení). Jde o parametr, který určuje váhu mezi predikovaným

stavem a měřením. Podle Kalmanova zisku jsou tedy jednotlivé složky stavového vektoru korigovány na základě reziduí. Hodnota odhadu současného stavu je kompromisem mezi hodnotou odhadu předchozího stavu a současným měřením. Kalmanův zisk se vypočte dle následujícího vztahu

$$K_k = p_k h / (h p_k h + r), \quad (2.37)$$

kde  $p$  je rozptyl predikce stavu a  $r$  značí rozptyl šumu měření. Update rozptylu predikce je spočten následně

$$p'_k = (1 - K_k h) p_k. \quad (2.38)$$

Pokud je rozptyl predikce nulový, je nulový také Kalmanův zisk, a tím není nutné odhad stavu korigovat měřenými hodnotami. Pokud ale má rozptyl predikce hodnotu 1 a rozptyl šumu měření je nulový, bude současný odhad stavu záviset pouze na současném měření. Čím vyšší bude hodnota rozptylu šumu měření, tím nižší bude Kalmanův zisk. V případě, kdy je hodnota šumu měření značně vysoká, není možné se na dané měření spoléhat.

Vztah (2.34) lze přepsat jako predikce stavu pomocí filtrace

$$\hat{x}_k = a \hat{x}'_{k-1}. \quad (2.39)$$

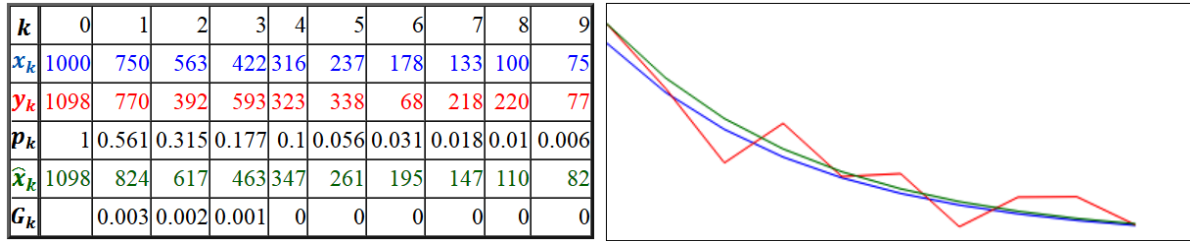
Relace (2.39) je predikce o tom, jaký by stav mohl být v dalším časovém kroku, zatímco rovnice (2.36) je filtrace (update) této predikce založená na měření. Rozptyl predikce rovněž závisí na své předchozí hodnotě a také obsahuje parametr  $a$

$$p_k = a p'_{k-1} a + q_k, \quad (2.40)$$

kde  $q$  představuje rozptyl šumu stavu. Rovnice (2.39) a (2.40) jsou součástí první fáze algoritmu KF nazvané predikce, zatímco rovnice (2.36) až (2.38) jsou součástí druhé fáze s názvem filtrace.

Následující obrázek zachycuje skutečný obecný stav (modrá křivka) a jeho měření (červená křivka) spolu s odhadem stavu (zelená křivka). Z tabulky na obrázku je patrné, že hodnoty rozptylu predikce a Kalmanova zisku jsou postupně s narůstajícím počtem časových kroků snižovány a hodnota odhadu stavu se blíží ke stavu skutečnému.





Obr. 15: Kalmanův filtr obecné funkce [43, upraveno]

Odhad polohy přijímače ale není jednodimenzionální záležitost, tudíž je nutné obecně pracovat s maticemi a vektory. Předchozí rovnice je tedy potřebné zapsat pomocí lineární algebry tak, že stavový model jsou následující rovnice (vektor měření je nyní označen jako  $\mathbf{Z}$ )

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k \quad (2.41)$$

$$\mathbf{Z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{z}_k. \quad (2.42)$$

Fáze predikce jsou rovnice

$$\hat{\mathbf{x}}_k = \mathbf{F}\hat{\mathbf{x}}_{k-1} \quad (2.43)$$

$$\mathbf{P}_k = \mathbf{F}\mathbf{P}'_{k-1}\mathbf{F}^T + \mathbf{Q}_k. \quad (2.44)$$

Fáze filtrace je soustava rovnic

$$\hat{\mathbf{x}}'_k = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{Z}_k - \mathbf{H}\hat{\mathbf{x}}_k) \quad (2.45)$$

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}^T(\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R})^{-1} \quad (2.46)$$

$$\mathbf{P}'_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k, \quad (2.47)$$

přičemž transponované matice jsou v relacích kvůli definici variance. Stavový vektor je označen jako  $\mathbf{X}$ . Matice  $\mathbf{P}$  je kovarianční matice chyb stavu, kdežto matice  $\mathbf{R}$  je kovarianční matice šumu měření. Matice  $\mathbf{I}$  značí jednotkovou matici. Matice  $\mathbf{Q}$  je kovarianční matice šumu procesu. Matice  $\mathbf{F}$  je tzv. matice dynamiky (přechodu), která umožní převést stav v kroku  $k-1$  na stav do kroku  $k$ . Matice  $\mathbf{H}$  je matice měření, která stav nijak neovlivňuje. V rovnici (2.45) je daná závorka nazývána vektorem inovace. Termín *rozptyl* je uvažován pro jednorozměrný model. Pro vícerozměrné modely je užit výraz *variance*, popř. *kovarianční matice*.

### Rozšířený Kalmanův filtr

Obecně lze říci, že pro určení pozice uživatele družicovým navigačním systémem GPS je tento algoritmus využíván častěji oproti Bancroftově a Newtonově metodě. Avšak standardní KF je schopen pracovat pouze s lineárními systémy, tudíž pro lokalizační účely je tento

algoritmus upraven na tzv. rozšířený Kalmanův filtr (EKF). Zobecnění je založeno na myšlence, kdy původní nelineární systém je linearizován v každém časovém kroku v okolí odhadu, střední hodnoty a kovariance. U rovnic modelu není opět uvažováno řízení. Relace pro stavový model vypadají následovně

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}) + \mathbf{w}_k \quad (2.48)$$

$$\mathbf{Z}_k = \mathbf{g}(\mathbf{x}_k) + \mathbf{z}_k. \quad (2.49)$$

Fáze predikce jsou vztahy

$$\hat{\mathbf{x}}_k = \mathbf{f}(\hat{\mathbf{x}}_{k-1}') \quad (2.50)$$

$$\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}'\mathbf{F}^T + \mathbf{Q}_k. \quad (2.51)$$

Fáze filtrace je soustava rovnic

$$\hat{\mathbf{x}}_k' = \hat{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{Z}_k - \mathbf{g}(\hat{\mathbf{x}}_k)) \quad (2.52)$$

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}^T(\mathbf{H}\mathbf{P}_k\mathbf{H}^T + \mathbf{R})^{-1} \quad (2.53)$$

$$\mathbf{P}_k' = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k, \quad (2.54)$$

kde  $\mathbf{f}$  a  $\mathbf{g}$  reprezentují nelineární vektorové funkce, zatímco  $\mathbf{F}$  a  $\mathbf{H}$  značí Jakobiány (matice parciálních derivací). Nelineární funkce jsou právě linearizovány jejich nahrazením Taylorova rozvoje pro nultý a první řád (lineární aproximace).

Výpočet nového odhadu stavu je prováděn v pravidelných intervalech na základě předchozího stavu a aktuálního měření. Ve fázi predikce je odhad stavu a priori a ve fázi filtrace je odhad stavu a posteriori. V první řadě je nutné před prvním krokem stanovit počáteční hodnotu stavového vektoru  $\mathbf{x}_0$  a počáteční hodnotu kovarianční matice predikce stavu  $\mathbf{P}_0$ . KF bývá velmi citlivý na hodnoty počátečních podmínek.

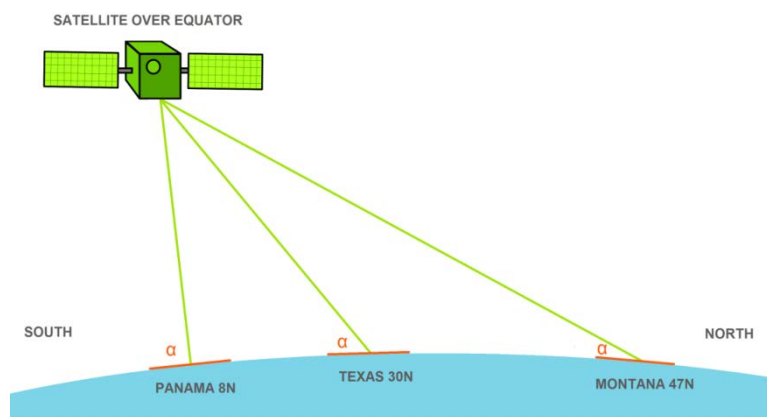
Ačkoli by se mohlo zdát, že Kalmanův filtr je iterativní algoritmus, není tomu tak. Hlavním rozdílem oproti předchozí metodě je, že Newtonův algoritmus provádí korekce iterativně v rámci jednoho časového okamžiku. Naopak Kalmanův filtr provede výpočet odhadu v jednom časovém okamžiku a použije tento výstup jako vstup pro další okamžik z důvodu zlepšení odhadu v následujícím okamžiku. Tato metoda byla sepsána a formálně upravena na základě analýzy z několika podkladů. [34, 42, 43, 44, 45, 46]

## 2.3 Matice vah

V praxi se lze u Bancroftovy a Newtonovy metody setkat také s váhovou maticí, která je ve vzorcích aplikována zejména v případech, kdy se liší kvalita jednotlivých měření pseudo-vzdálenosti. Matice vah poté umožňuje zvýšení přesnosti odhadu polohy.

V předchozí části této kapitoly bylo vysvětleno, že značný vliv na signál má ionosféra. Přijímač zpracovává signály od družic, jejichž elevační úhel (úhel mezi horizontem a družicí) je různý. Doba přenosu signálu u družic s malým elevačním úhlem je delší než u družic s vyšší elevací. Signál je tedy více ovlivněn ionosférou. Družice s malými elevačními úhly by tedy při výpočtu měly mít menší váhu než družice s vyššími elevačními úhly. Názorně je tento problém zachycen na obrázku č. 16, kde písmeno  $\alpha$  značí elevační úhel. [47]

Dalším případem aplikace váhové matice je využití multikonstelačního přijímače, tedy přijímače schopného zpracovávat signály z více systémů GNSS současně. Komplikací je, že systém GPS je zatížen jinými systémovými chybami než systém GLONASS. V takovém případě je využití váhové matice důvodem pro odlišení signálů různých GNSS systémů při výpočtech, a tím zvýšení přesnosti odhadu polohy. [47]



Obr. 16: Různé elevační úhly družice [14, upraveno]

Matice vah  $W$  jednotlivých měření je diagonální matice obsahující inverzní hodnoty rozptýlů odhadovaných pseudovzdáleností. Obecně se pro zjednodušení předpokládá vzájemná nezávislost daných měření, tudíž nulová korelace mezi nimi. Proto jsou ostatní prvky mimo diagonálu nulové. Dále se předpokládá Gaussovo normální rozdělení. Pro jednodušší zápis je níže uvedena váhová matice pro čtyři měření jako inverzní

$$\mathbf{W}^{-1} = \begin{pmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix}. \quad (2.55)$$

Pokud je brána v úvahu matice vah pro výpočet polohy, měly by být vztahy (2.18) u Bancroftovy metody a (2.33) u Newtonovy metody modifikovány tak, jak je uvedeno níže

$$\mathbf{s}_r = (\mathbf{B}^T \mathbf{W} \mathbf{B})^{-1} \mathbf{B}^T \mathbf{W} (\mathbf{a} + \Delta \mathbf{e}) \quad (2.56)$$

$$\Delta \mathbf{s} = (\mathbf{A}^T \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{W} \mathbf{L}. \quad (2.57)$$

Avšak pro zjednodušení je v této práci uvažováno, že hodnoty rozptylů odhadovaných pseudovzdáleností jsou stejné, tudíž matematicky lze rozptyl vytknout před maticí, čímž se z matice vah stane matice jednotková násobená rozptylem. Jinými slovy, je předpokládáno, že chyba odhadu měření má normální rozdělení se střední hodnotou nula a rozptylem daným hodnotou  $\sigma^2$  násobenou jednotkovou maticí, což je uvedeno již na začátku podkapitoly 2.2. Díky tomuto předpokladu není u Bancroftovy a Newtonovy metody matice vah uvažována. [27, 33]

## 2.4 Faktor DOP

Jednoznačným parametrem kvality určení polohy je bezrozměrný faktor DOP (angl. Dilution of Precision). Základem je výpočet, který uvažuje relativní polohu každé družice vzhledem k přijímači a zároveň k ostatním družicím. DOP tedy závisí pouze na konstelaci družic. Vyšší hodnota odráží nepříliš vhodné rozmístění, ze kterého následně nelze vypočítat polohu s relativně velkou přesností. Čím blíže jsou družice vůči sobě umístěny, tím horší výsledky přesnosti jsou uživateli poskytovány. Ideální situace pro měření ke čtyřem družicím nastává, pokud je jedna družice nad přijímačem (v nadhlavníku) a zbylé tři jsou rozmístěny kolem  $20^\circ$  nad obzorem a  $120^\circ$  vůči sobě. Tento předpoklad je ale například ve městech špatně dosažitelný. S každou další družicí je hodnota DOP snižována. [1]

Činitelé DOP se vypočítávají ze čtvercové matice  $(\mathbf{A}^T \mathbf{A})^{-1}$ , přičemž  $\mathbf{A}$  je matice směrových kosinů

$$(\mathbf{A}^T \mathbf{A})^{-1} = \begin{pmatrix} D_{xx} & D_{yx} & D_{zx} & D_{tx} \\ D_{xy} & D_{yy} & D_{zy} & D_{ty} \\ D_{xz} & D_{yz} & D_{zz} & D_{tz} \\ D_{xt} & D_{yt} & D_{zt} & D_{tt} \end{pmatrix}. \quad (2.58)$$

Jednotlivými složkami DOP jsou HDOP (snížení horizontální přesnosti), VDOP (snížení vertikální přesnosti), PDOP (snížení polohové přesnosti), TDOP (snížení přesnosti časových měření) a GDOP (snížení geometrické přesnosti). Hodnoty HDOP a VDOP jsou stanoveny následně

$$HDOP = \sqrt{D_{xx} + D_{yy}} \quad (2.59)$$

$$VDOP = \sqrt{D_{zz}} \quad (2.60)$$

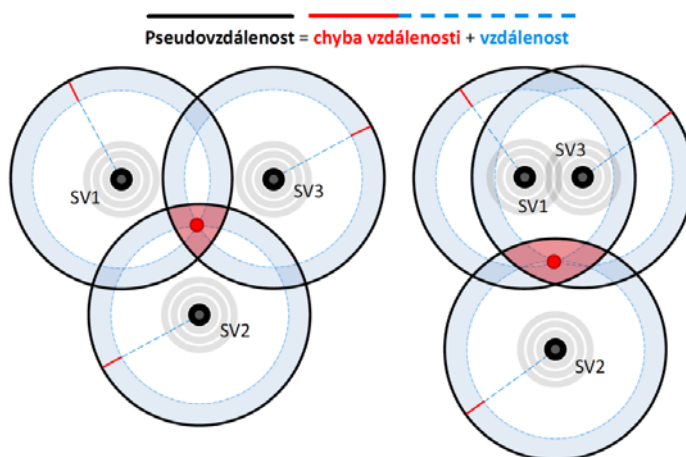
$$TDOP = \sqrt{D_{tt}} \quad (2.61)$$

Hodnoty PDOP a GDOP jsou určeny jako

$$PDOP = \sqrt{HDOP^2 + VDOP^2} \quad (2.62)$$

$$GDOP = \sqrt{PDOP^2 + TDOP^2}. \quad (2.63)$$

Hodnota GDOP je tedy určena jako druhá odmocnina stopy matice (2.58). Příjímač vypočítává každou složku DOP pro všechny možné kombinace dostupných družic a pro stanovení polohy použije tu skupinu družic, pro kterou vychází hodnota parametru PDOP nejnižší. Obrázek č. 17 zachycuje vliv geometrického uspořádání třech družic na přesnost odhadu polohy. Jsou-li družice optimálně rozmístěné, je plocha, na které se přijímač vyskytuje, menší než u družic nevhodně rozmístěných. Čím větší je tato plocha, tím méně přesný je odhad polohy. Optimální hodnota PDOP je v rozmezí 1 - 4. Hodnota 6 a více již značí nevhodné rozmístění dostupných družic a není tedy zaručena dostatečná přesnost určení polohy. Samozřejmě ale vždy záleží na uživateli, jakou přesnost vyžaduje. Obvykle je v přijímači umožněno nastavit limit pro hodnotu PDOP, po jehož překročení je daná konstelace družic při výpočtu ignorována. [1, 47]



Obr. 17: Vliv rozmístění družic na přesnost odhadu polohy [2]

### 3 Analýza metod určení odhadu polohy

V předchozí kapitole je popsán postup výpočtu odhadu polohy uživatele pro tři matematické algoritmy, které bývají v GPS přijímačích implementovány nejčastěji. Ovšem význam této práce spočívá zejména v posouzení kvality, tedy přesnosti, poskytnutého odhadu získaného právě těmito metodami, a to také v souvislosti s reakcí na přítomnost poruchy v měření.

Testování kvality by pravděpodobně bylo možné provést analyticky, avšak šlo by o výpočetně a časově velmi náročný proces. Jako vhodnější varianta se tedy jeví testování na základě simulací pomocí výpočetního programu. V případě takového testování obecně platí, že pro skutečně relevantní výsledky je nezbytné pracovat s relativně velkým vzorkem statistických dat. Proto nestačí stanovit odhad pouze jednou, ale bude nutné provést několik set až tisíc výpočtů odhadu.

Simulace bude provedena ve výpočetním programu pro každou metodu, a to v závislosti na kombinaci hodnot velikosti injektované poruchy v měření a velikosti směrodatné odchylky (STD) šumu. Pro dané kombinace bude vždy provedeno několik set až tisíc výpočtů odhadu polohy právě z důvodu získání relevantních výsledků. Hodnota STD šumu bude mít pochopitelně vliv na výslednou přesnost, ovšem neměla by v případě nepřítomnosti poruchy značně ovlivnit procentuální náležitost vůči kovariančnímu elipsoidu (vysvětleno níže). V rámci testování vlivu poruch na dané metody bude tedy vhodné provést toto testování vlivu poruch také v závislosti na hodnotě STD šumu.

Dané simulace by měly ověřit následující hypotézy. V případě, kdy bude generována relativně malá chyba spolu s nadefinovanou poměrně malou hodnotou STD šumu, by měl být odhad polohy ovlivněn téměř stejně jako v případě, kdy bude jak hodnota injektované chyby, tak STD mnohem vyšší. Bude-li ovšem generovaná porucha značně vyšší vůči hodnotě STD, bude mít tato kombinace mnohem větší dopad na odhad polohy, než v případě vysokých hodnot poruchy a STD. Toto tvrzení má logické vysvětlení. Pokud je totiž velikost generované chyby vysoká, přesto téměř shodná jako hodnota STD, nastává případ, kdy se porucha v šumu částečně „ukryje“. Právě z tohoto důvodu budou simulace prováděny pro různé hodnoty STD.

Další hypotéza vyplývá z vlastností daných metod. Jak Bancroftova, tak Newtonova metoda by měla reagovat na přítomnost poruchy v měření prakticky okamžitě. Ovšem algoritmus Kalmanova filtru pracuje s pamětí, tudíž by na poruchu měl reagovat s určitým zpožděním. Totéž platí pro případ, kdy porucha zmizí. Na rozdíl od předchozích dvou metod by měl

Kalmanův filtr počítat s poruchou ještě určitou dobu po jejím odeznění. Toto ale záleží na době působení poruchy a také na její velikosti. V případě, kdy bude porucha přítomna pouze pár sekund, nemusí se ve výpočtu odhadu polohy dle Kalmanova filtru projevit. Tato hypotéza bude rovněž simulačně ověřena, nebo vyvrácena.

Výstupem těchto simulací bude grafické znázornění chyb odhadu polohy pro jednotlivé metody v závislosti na časovém okamžiku, kde by měla být zachycena reakce na přítomnost poruchy. V předchozí kapitole je uvedeno, že přesnost odhadu polohy je značně ovlivněna hodnotou PDOP. Tudíž budou také graficky zachyceny velikosti chyb odhadu polohy, jež budou klasifikovány právě dle hodnoty PDOP. Chyby odhadu polohy těchto metod budou samozřejmě porovnány a zobrazeny v také tabulkách. U Newtonovy metody a Kalmanova filtru budou rovněž posouzeny kovarianční matice odhadů. Zdali je odhad těchto dvou metod správný, bude ověřeno na základě vyhodnocení kovariančního elipsoidu, který je popsán níže. Ovšem při injektaci poruchy se očekává, že nastanou jisté problémy, a to z důvodu nedodržení počátečních podmínek.

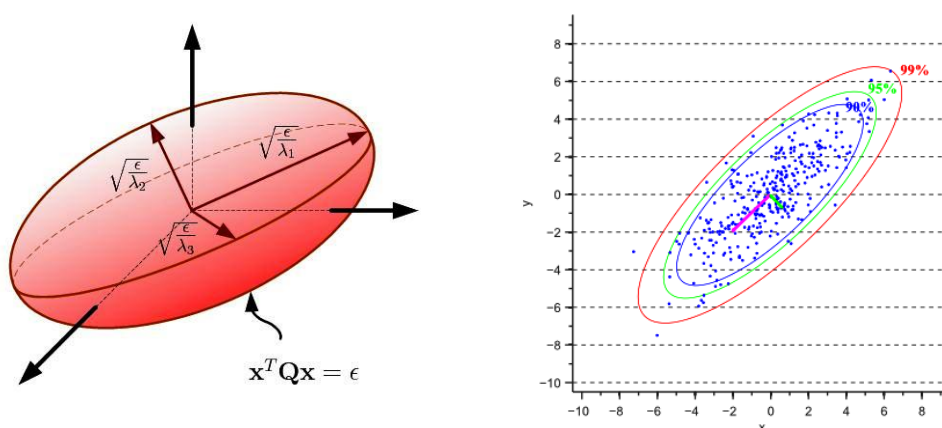
Díky výstupům bude možné provést analýzu přesnosti daných metod a zhodnotit, jak tyto metody reagují na různé hodnoty poruch v závislosti na odlišných hodnotách STD šumu. Smyslem těchto simulací je tedy ověření výše uvedených předpokladů.

Pro ověření správnosti odhadu polohy dle Newtonovy metody a Kalmanova filtru je využíván tzv. kovarianční elipsoid. Jde o prostorový útvar, jenž charakterizuje daný odhad. Středem elipsoidu je výsledný odhad polohy, přičemž roztažení a natočení elipsoidu je dáno kovarianční maticí odhadu. Velikosti os elipsoidu jsou určeny vlastními čísly kovarianční matice, kdežto natočení elipsoidu v prostoru je dáno vlastními vektory. Elipsoid tedy stanovuje jakousi důvěru v daný odhad. Princip ověření správnosti odhadu spočívá v určení polohy ověřovaného bodu relativně vůči danému elipsoidu. Ověřovaným bodem je v tomto případě skutečná poloha přijímače, která je známa. Pokud leží skutečná poloha uvnitř nebo na hraně elipsoidu pro různé procentuální hodnoty hladiny významnosti (vysvětleno níže), dá se předpokládat, že je odhad správný. Termín *správný* je v těchto případech založen na pravděpodobnosti, tedy pokud pro určité procento odhadů, charakterizovaných kovariančním elipsoidem, platí, že skutečná poloha leží v tomto elipsoidu, považuje se odhad za správný.

Při práci s kovariančním elipsoidem jsou využívány tzv. hladiny významnosti, které souvisí s pravidlem tří sigma. To říká, že by se měly téměř všechny relevantní hodnoty statistického souboru nacházet do tří směrodatných odchylek od průměru při normálním rozdělení. Dle

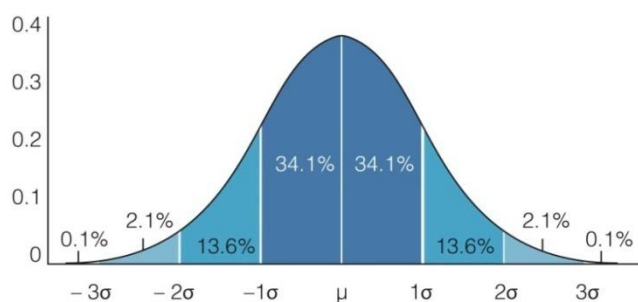
toho, jak velká část je do souboru zahrnuta a co je považováno za zanedbatelně výjimečné hodnoty, se určí  $1\sigma$  (68% hladina významnosti),  $2\sigma$  (95% hladina významnosti), nebo  $3\sigma$  (99% hladina významnosti). Vyřazené hodnoty se považují za hrubé chyby. Hodnoty 68 % a 95 % jsou využity při vyhodnocení v této simulaci. V simulacích půjde tedy o vyhodnocení dopadu konkrétních poruch na velikost chyby odhadu polohy pro různé hodnoty STD, tedy posouzení jak se změní procenta náležitosti do kovariančního elipsoidu při různých hodnotách STD šumu a poruch. [42]

Kovarianční elipsoid je ilustrován na obrázku č. 18 spolu s kovarianční elipsou pro hodnoty hladiny významnosti 90 %, 95 % a 99 %.



Obr. 18: Kovarianční elipsoid a kovarianční elipsa [48, 49]

Graf na obrázku č. 19 zachycuje normální rozdělení náhodné veličiny včetně vymezených hranic pro hladiny významnosti.



Obr. 19: Normální rozdělení náhodné veličiny s vyznačenými hladinami významnosti [50]

### 3.1 Příprava pro simulační analýzu

Simulace pro analýzu přesnosti odhadu polohy využívá data poskytnuta společností AŽD Praha, s.r.o. Tato data mimo jiné obsahují informace o skutečné poloze přijímače, o počtu



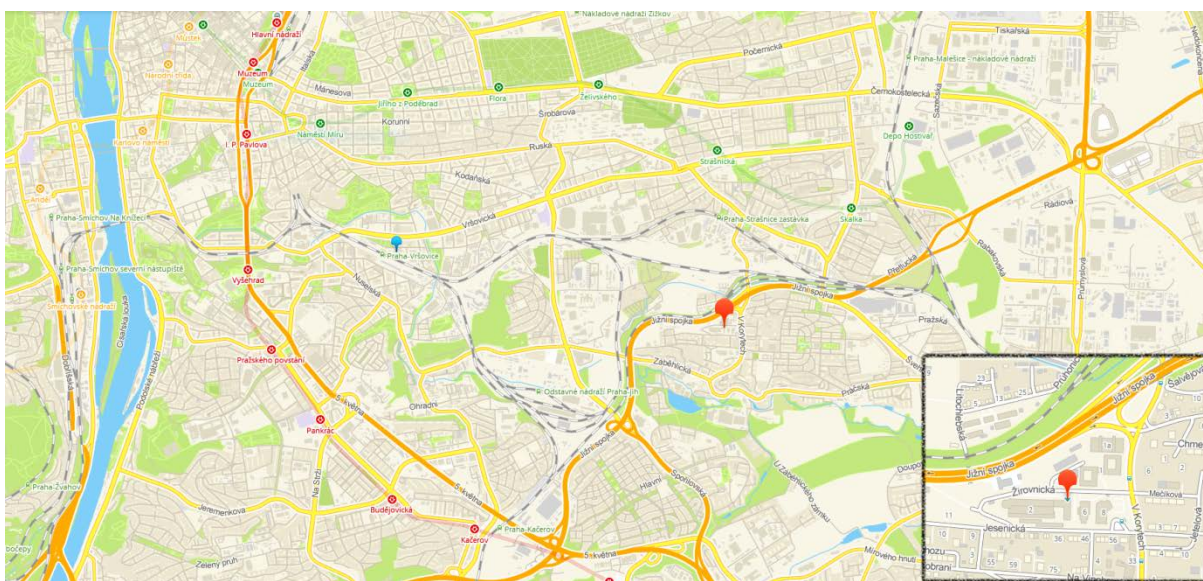
a pozicích všech družic včetně příslušných elevací, pseudovzdáleností a časových rozdílů družicových hodin vůči času GPST.

### Data pro simulační analýzu

Data byla vygenerována, dle předem definovaného modelu, pro efemeridy družic ze dne 18. ledna 2016, a to pro 43 082 časových okamžiků s periodou 2 sekundy. Tudíž celkově jde o data pro celý střední sluneční den bez 236 sekund (tedy 86 164 s). Toto je záměrné, poněvadž konstelace družic systému GPS se vždy opakuje po jednom hvězdném (siderickém) dni, jenž trvá právě 86 164 sekund.

Poloha přijímače a pozice družic jsou uvedeny v kartézském souřadném systému. Družice jsou k jednotlivým časovým okamžikům přiřazeny dle určitých pravidel. Družice s elevačními úhly do  $6^\circ$  nejsou do daného okamžiku zahrnuty, naopak družice s elevací nad  $70^\circ$  jsou zahrnuty všechny. Ty, jejichž elevace jsou v rozmezí  $6^\circ - 70^\circ$  jsou vybrány náhodně, a to s ohledem na kritérium, které říká, že každému časovému okamžiku musí přiřazeno nejméně pět, ale ne více než deset družic.

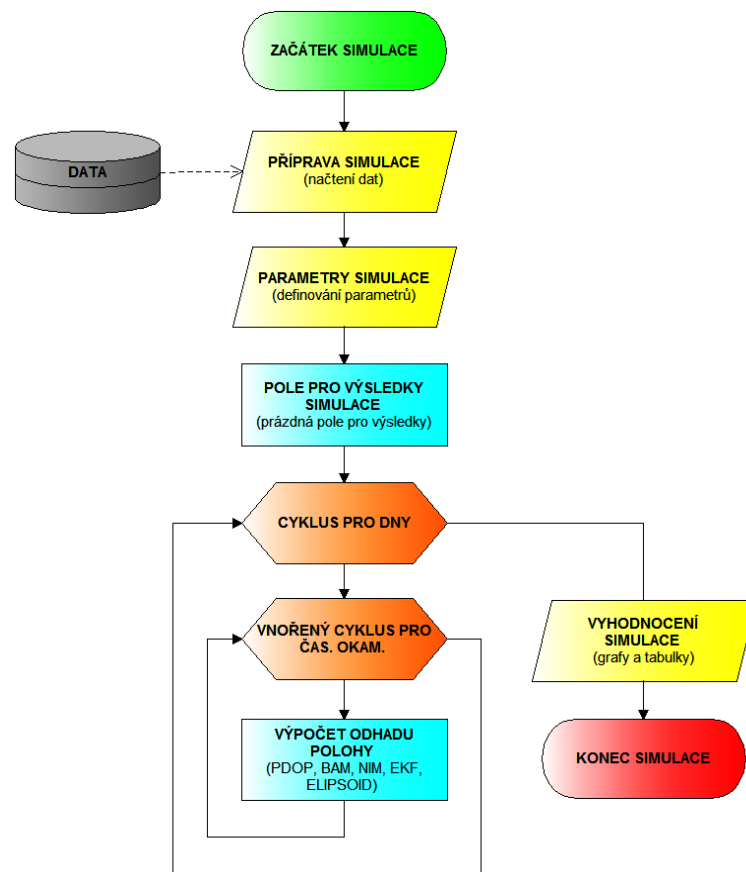
Vzhledem k tomu, že je posuzována přesnost odhadu polohy, je nezbytné znát skutečnou polohu přijímače, vůči které budou vypočtené odhady relativně vztaženy. Skutečná poloha je statická, tudíž pro všechny časové okamžiky stejná. Touto referenční polohou přijímače byla zvolena střecha hlavního sídla společnosti AŽD Praha s.r.o. (Žirovnická 2/3146, 106 17, Praha 10) se souřadnicemi v kartézském souřadném systému  $X = 3\,972\,541,6$ ,  $Y = 1\,026\,722,1$  a  $Z = 4\,867\,206,3$ . Poloha je vyznačena na níže uvedeném obrázku.



Obr. 20: Zobrazení referenční polohy přijímače [51]

Pro simulační ověření daných algoritmů byl využit výpočetní software MATLAB verze R2016b se studentskou placenou licencí spolu s doplňkovým nástrojem pro statistické účely „Statistics Toolbox“. V tomto programu byly jednotlivé algoritmy vytvořeny do skriptů tak, aby vše bylo v souladu s textem této práce (tj. postupy, značení proměnných a podobně). Ke každému řádku skriptu je uveden komentář, který stručně a jednoduše popisuje chování příslušného příkazu. Není tedy nutné, aby tato kapitola rozebírala skripty řádek po řádku, ovšem je nezbytné nastínit a objasnit průběh celé simulace.

Všechny příslušné skripty programu MATLAB spolu s komentáři a výstupy jsou uvedeny jak v příloze, tak na přiloženém CD, přičemž vše bylo vytvořeno autorem této práce. Následující obrázek zachycuje zjednodušený vývojový diagram celého simulačního procesu pro výpočet odhadu polohy včetně jeho posouzení.



Obr. 21: Vývojový diagram průběhu simulace [vlastní tvorba]

Pro výpočet odhadu polohy bylo v programu MATLAB vytvořeno celkem sedm separátních souborů. Šest z nich reprezentuje dané algoritmy a jiné funkce, které jsou vyvolány z hlavního skriptu s názvem „Skript\_Algorithmus.m“. Soubory s funkcemi byly pojmenovány následovně

- Funkce\_PDOP.m
- Funkce\_Bancroft.m
- Funkce\_Newton.m
- Funkce\_KalmanR2.m
- Funkce\_KalmanR3.m
- Funkce\_Elipsoid.m,

přičemž v těchto souborech probíhají výpočty odhadu polohy spolu se stanovením dalších potřebných údajů, které jsou poté v hlavním skriptu zpracovány, případně vyhodnoceny.

### **3.1.1 Skript\_Algoritmus.m**

Hlavní skript je rozdělen na několik sekcí, které jsou uvedeny jako samostatné bloky ve vývojovém diagramu na obrázku č. 21 a rovněž jsou níže stručně popsány.

#### **Příprava simulace**

Před samotným spuštěním simulačního procesu je nezbytné vymazat výsledky a proměnné z předchozí simulace. K tomuto účelu slouží mimo jiné první sekce hlavního skriptu. Rovněž zde dochází k načtení dat vygenerovaných, případně získaných z reálného měření. Kvůli poměrně velkému počtu vygenerovaných časových okamžiků by byla simulace časově náročná, proto byla vybrána pouze část z celého souboru časových okamžiků, a to tak, aby přesto byl splněn požadavek na dostatečně vypovídající vzorek dat. Nicméně veškeré skripty jsou vytvořeny pro možnost zpracování celého souboru dat.

Na úplném začátku této sekce je příkaz, který zobrazí skutečné aktuální datum a čas začátku simulace. Na konci celého skriptu je uveden tentýž příkaz, aby bylo možné odvodit dobu průběhu celé simulace. Také je v této sekci nastaven časovač, čili od této chvíle bude počítán uplynulý čas simulačního procesu v sekundách, který bude v průběhu a na konci zobrazován.

#### **Parametry simulace**

Pro správný chod simulace je nezbytné nadefinovat vstupní parametry. Především jde o počet časových okamžiků. Jak je již uvedeno v předchozím textu, je žádoucí stanovit odhad polohy pro několik tisíc okamžiků. Data sice byla vygenerována pro 43 083 okamžiků (včetně okamžiku pro čas 0 sekund) v rámci jednoho dne, nicméně, jak již bylo uvedeno, nebude simulace provedena pro všechny okamžiky. Dalším parametrem je počet dnů, po které bude měření simulačně probíhat.

U veličiny šumu se předpokládá normální rozdělení, které je definováno nulovou střední hodnotou a směrodatnou odchylkou o určité velikosti. Právě tato hodnota STD bude měněna.

V této sekci je také uvedena počáteční podmínka pro Newtonovu metodu a Kalmanův filtr. Daná podmínka, jež je záměrně v této sekci pro případnou modifikaci bez nutnosti zásahu do funkcí, představuje střed náměstí obce České Budějovice se souřadnicemi v kartézském souřadném systému  $X = 4\,061\,678,8$ ,  $Y = 1\,048\,478,1$  a  $Z = 4\,789\,000,1$ .

V praxi je možné stanovit počáteční podmínku různými způsoby. Před samotným výpočtem odhadu mohou být nejdříve provedeny výpočty hodnoty DOP. Jakmile bude DOP nižší než předem stanovená hodnota, použije se příslušná kombinace družic (pro niž platí tato hodnota DOP) pro výpočet polohy, která bude vstupovat jako počáteční podmínka pro vlastní stanovení odhadu. Při aplikaci do železniční dopravy se obvykle předpokládá znalost lokality, ve které bude operovat drážní vozidlo s přijímačem GPS. Od toho se také odvíjí stanovení počáteční podmínky. Pokud je známa konkrétní trať, bude považován za počáteční podmínku například přibližný střed této trati.

Rovněž je možné provést před hlavním výpočtem odhad polohy podle Bancroftovy metody. Výstupní odhad této metody může sloužit právě jako počáteční podmínka pro Newtonovu metodu či pro Kalmanův filtr. Jednou z možností je také použít pro počáteční podmínku nulové hodnoty.

### **Pole pro výsledky simulace**

Pro uložení a následné zobrazení výsledných hodnot jak odhadů polohy, tak souvisejících výpočtů je vhodné nejdříve vytvořit prázdná maticová pole, do kterých budou výsledky výpočtů zapisovány. Tato maticová pole jsou nadefinována s defaultními hodnotami „NaN“ (angl. Not a Number), aby po skončení simulace bylo zřejmé, že jsou tyto hodnoty skutečně přepsány. V případě, kdyby byla před začátkem simulace v buňce pro výsledek hodnota nula, nebylo by po provedení výpočtů zjevné, zdali jde o původní hodnotu, nebo nulový výsledek.

Vytvoření této sekce je efektivní zejména z výpočetního hlediska. Díky tomu, že jsou předem nadefinována prázdná pole pro výsledné hodnoty, nebude simulace probíhat způsobem, kdy se v prvním cyklu vytvoří buňka s jedním výsledkem, v druhém cyklu se předchozí buňka smaže a vytvoří se dvě buňky pro výsledky. V dalším cyklu by byly tyto buňky smazány a nahrazeny třemi buňkami. Daný způsob by byl časově mnohem náročnější.

Výsledkem budou velikosti chyb odhadu polohy celkově ve všech třech osách X, Y a Z, ale také chyby daných matematických metod v jednotlivých osách separátně.

## Cyklus pro dny

Jak je již patrné z diagramu na obrázku č. 21, proces simulace je nastaven tak, aby proběhl nejdříve výpočet odhadu polohy pro časové okamžiky v rámci jednoho dne. Teprve poté proběhne přechod do následujícího dne. Počet dní lze regulovat v sekci „Parametry simulace“, nicméně v této práci budou veškeré výpočty probíhat po dobu 365 dní, tedy po celý rok.

Na konci každého dne bude zobrazena v průběhu simulace informace o pořadí simulačního dne včetně uplynulého času od začátku pro ujištění, zdali simulace stále pracuje.

## Vnořený cyklus pro časové okamžiky

Počet časových okamžiků je pevně dán vygenerovanými daty. Časový okamžik  $t = 1$  je nultá sekunda dne, časový okamžik  $t = 3$  je čtvrtá sekunda dne atd. Ačkoli ve druhé kapitole nebyla pro zjednodušení uvažována časová diference hodin družice a GPST, v simulacích se s ní pracuje z důvodu přiblížení se skutečnosti. Veličiny „Pozice družic“, „PDOP“ a „Poloha přijímače“ jsou v této simulaci pro všechny dny stejné, poněvadž nejsou na konkrétním dni závislé. Jsou závislé, kromě statické polohy přijímače, pouze na časovém okamžiku.

V simulaci je uměle generován šum, jenž je přiřazený k teoreticky určené hodnotě pseudovzdálenosti, která byla vygenerována modelem pro každou družici. Šum je v simulaci záměrně přidán z důvodu toho, že ve skutečnosti je šum součástí naměřené pseudovzdálenosti.

Pro vyhodnocení chyby odhadu je ve skriptu použita Euklidova norma, což je odmocnina ze součtu druhých mocnin všech prvků rozdílu skutečné polohy přijímače a odhadu. U veličiny „Stopa matice“ je záměrně uvedena odmocnina, kvůli kovariančnímu elipsoidu. Cílem je si jedním číslem charakterizovat kovarianční elipsoid (jeho osy). Stopa matice je rovna součtu jejích diagonálních prvků, ale také součtu vlastních čísel. Právě odmocnina vlastních čísel kovarianční matice charakterizuje velikosti jednotlivých os kovariančního elipsoidu.

### 3.1.2 Funkce\_PDOP.m

Funkce pro výpočet hodnoty PDOP je definována dle kapitoly 2.4. Základem je stanovení matice směrových kosinů a její následné zpracování. Součástí funkce je podmínka, týkající se hodnoty matice. Inverzi matice obecně lze provést pouze v případě, je-li původní matice čtvercová. Navíc musí mít plnou hodnotu, proto je požadavek na hodnotu rovný čtyřem.

V praxi se ovšem hodnota PDOP počítá odlišným způsobem, protože není známa skutečná poloha přijímače. Pak ale neplatí, že by hodnoty PDOP byly pro všechny dny stejné. Vzhledem k tomu, že se předpokládá pro každý den stejná konstelace družic a poloha přijímače

je statická, je vytvořeno pole pro výsledné hodnoty PDOP v závislosti na časovém okamžiku pouze pro jeden den.

### 3.1.3 Funkce `Bancroft.m`

Vstupem pro výpočet odhadu polohy dle Bancrofta jsou pozice družic s příslušnými pseudovzdálenostmi. Výstupem bude pouze daný odhad. Celý postup je popsán v kapitole 2.2.1, přičemž výsledný odhad je vybrán ze dvou kořenů kvadratické rovnice na základě toho, kterému odhadu přísluší nižší hodnoty reziduí.

### 3.1.4 Funkce `Newton.m`

Na vstupu Newtonovy metody je nezbytné, vyjma pozic družic a jejich pseudovzdáleností, stanovit počáteční podmínku pro výpočet výsledního odhadu. V této simulaci jsou podmínkou souřadnice polohy náměstí v obci České Budějovice.

Počáteční poloha je v Newtonově metodě iteračním procesem korigována až do chvíle, dokud se nebude hodnota odhadu příliš lišit od předchozí hodnoty. Jinými slovy, pokud bude vektor korekcí souřadnic a časové difference pod předem stanovenou hodnotou tolerance, bude výstup považován za výsledný odhad polohy. Ve funkci je stanovena navíc podmínka pro maximální počet iterací proto, aby neprobíhaly výpočty do nekonečna. Ovšem v případě, kdy bude dosaženo maximálního počtu iterací, není možné brát výstup za výsledný odhad, a tudíž bude na výstupu zobrazena hodnota NaN.

Výstupem této metody je odhad polohy a jeho kovarianční matice, jež je v hlavním skriptu nahrazena metrikou pro jednodušší porovnání, konkrétně stopou a determinantem. Byl dodržen formální zápis funkce dle kapitoly 2.2.2.

### 3.1.5 Funkce `KalmanR2.m` a Funkce `KalmanR3.m`

Výpočet odhadu pomocí Kalmanova filtru, vytvořeného dle kapitoly 2.2.3, vyžaduje více vstupních veličin. Opět je nutné znát pozice družic a příslušné zdánlivé vzdálenosti spolu s počáteční podmínkou. Ta je rovněž stanovena pro České Budějovice. U Kalmanova filtru je také stanovena počáteční podmínka pro kovarianci dané počáteční podmínky pro polohu.

Jak již ale bylo napsáno, Kalmanův filtr provádí korekci počáteční podmínky v rámci celého dne, nikoli v rámci jednoho okamžiku, jako je tomu u Newtonovy metody. Například tedy v časovém okamžiku  $n$  je provedena korekce podle hodnoty v čase  $n-1$ , což je predikce pro časový okamžik  $n$ . V čase  $n$  je dále tato predikce opravena v závislosti na hodnotách

aktuálního měření v čase  $n$ , což se nazývá filtrace. Z filtrace je dále provedena predikce, ale již pro další časový okamžik.

Kvůli programátorské realizaci simulace nelze v případě začátku nového dne brát hodnoty ze dne předchozího, tudíž za počáteční podmínku pro nový den je považována opět poloha obce České Budějovice. Ve skriptu je toto ošetřeno tím, že počáteční podmínka pro Kalmanův filtr je uvedena v cyklu pro dny. Výstupem této metody je opět odhad polohy spolu s jeho kovarianční maticí.

### **Řád modelu parametrů**

Byly vytvořeny dvě funkce pro Kalmanův filtr, které se odlišují řádem modelu parametrů, avšak princip a formální zápis obou verzí je totožný. Model v podstatě říká, jak se stav vyvíjí, přičemž řádem modelu se rozumí počet stavových proměnných (parametrů) modelu.

Analyzován bude druhý a třetí řád (R2 a R3). Model druhého řádu obsahuje souřadnice polohy přijímače včetně biasu a rovněž rychlost ve všech třech směrech. Model třetího řádu obsahuje navíc zrychlení ve všech třech směrech. Kromě rozměrů příslušných matic mají oba řády rozdílně definované kovarianční matice šumu stavu a matice dynamiky. Tyto matice byly vytvořeny dle zdroje [19].

U kovarianční matice šumu stavu se vyskytuje tzv. spektrální výkonová hustota, která má pro bias několikanásobně nižší hodnotu než pro souřadnice polohy, a to z důvodu relativně vysoké stability krystalu hodin.

### **3.1.6 Funkce\_Elipsoid.m**

Funkce pro ověření správnosti odhadu byla vytvořena dle úvodního textu kapitoly 3. Pracuje s hladinami významnosti 68 % a 95 %. Výstupem je logická jednička v případě, kdy je odhad správný a logická nula pro odhad vyhodnocený jako nesprávný.

## **3.2 Průběh a vyhodnocení simulace**

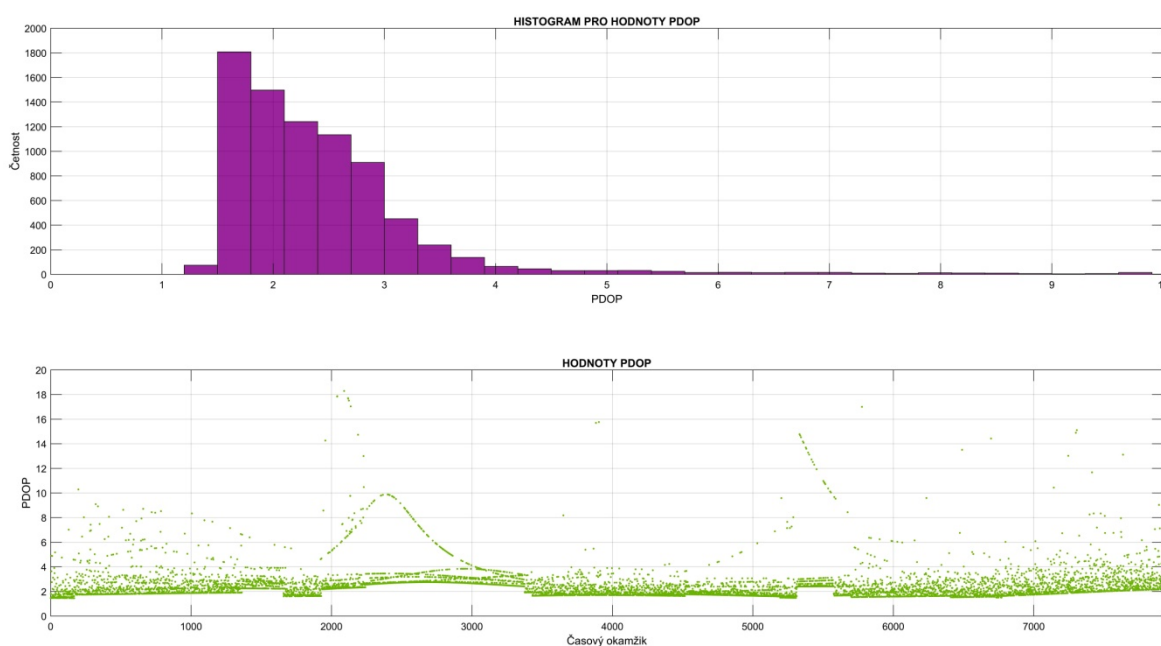
Pro zhodnocení přesnosti a reakce na poruchy u metod určení odhadu polohy bylo provedeno celkem osm simulací, jež se lišily kombinacemi velikostí injektované poruchy a parametru šumu (směrodatná odchylka šumu). Všechny simulace probíhaly po dobu jednoho simulačního roku, přičemž v rámci jednoho dne byl proveden výpočet pro 8 000 časových okamžiků s periodou 2 s, což odpovídá necelým 4,5 hodinám simulačního měření denně. Pro posouzení přesnosti daných metod je tento počet časových okamžiků dostačující. Všechny grafy (včetně

neuvedených v této kapitole) jsou součástí přílohy a přiloženého CD. Níže je zobrazena tabulka s parametry jednotlivých simulací spolu s dobou jejich průběhu.

Tab. 1: Tabulka parametrů simulace

Simulace	STD šumu [m]	Porucha [m]	Počet čas. okam.	Počet dnů	Doba simulace [h:mm:ss]
1	2,5	0	8000	365	2:16:03
2	10	0	8000	365	2:33:58
3	2,5	2,5	8000	365	2:00:10
4	2,5	10	8000	365	2:20:46
5	10	2,5	8000	365	2:39:44
6	10	10	8000	365	1:33:51
7	2,5	50 (6 s)	8000	365	1:34:27
8	2,5	50 (60 s)	8000	365	1:44:15

Porucha byla injektována do dat družice s číslem 29, protože se vyskytuje v oněch 8 000 okamžicích simulace téměř po celou dobu. Zároveň bude možné sledovat reakci daných metod i po odeznění injektované poruchy, protože tato družice nebude dostupná od okamžiku zhruba 6 900. Kromě simulací s relativně malou a velkou poruchou byly provedeny ještě dvě simulace s extrémní poruchou 50 m, a to po dobu 6 a 60 sekund. Díky tomu bude také možné posoudit reakci daných metod na kratší a delší působení značně velké poruchy. Všechny simulace mají shodné vypočtené hodnoty PDOP a pozice družic. Na obrázku č. 22 je zachycen histogram hodnot PDOP včetně jejich časového průběhu.

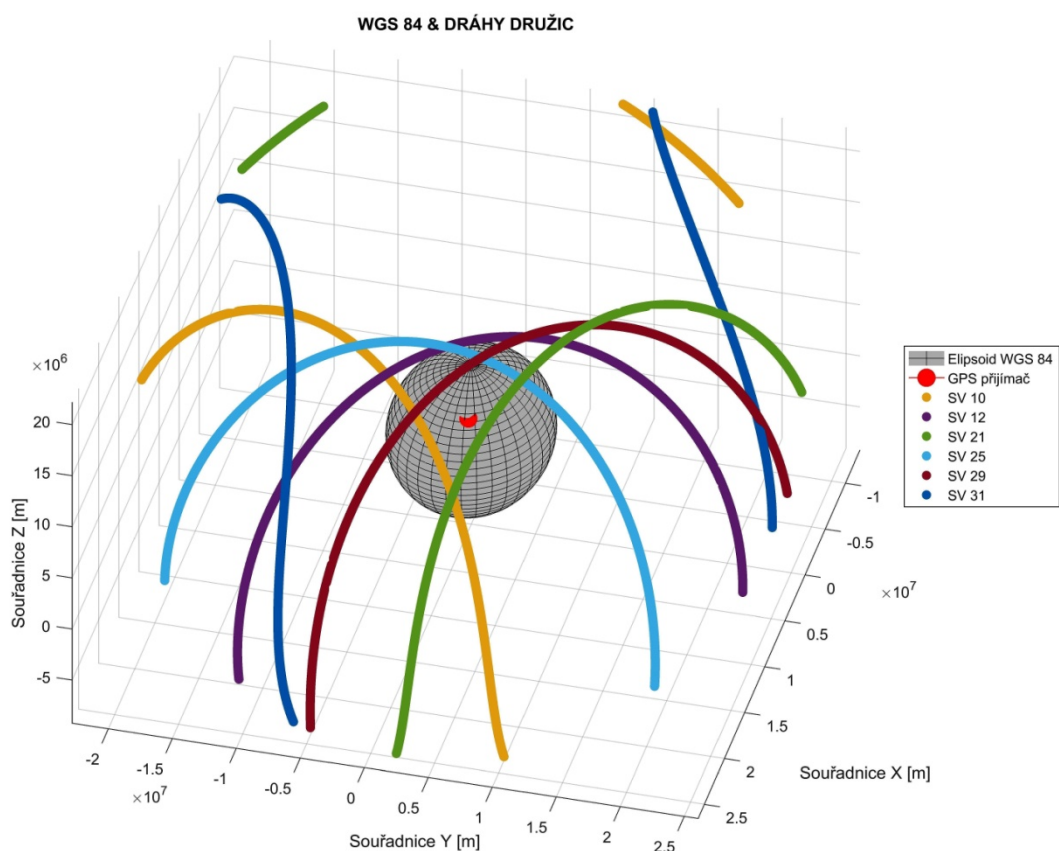


Obr. 22: Histogram hodnot PDOP a hodnoty PDOP dle časových okamžiků



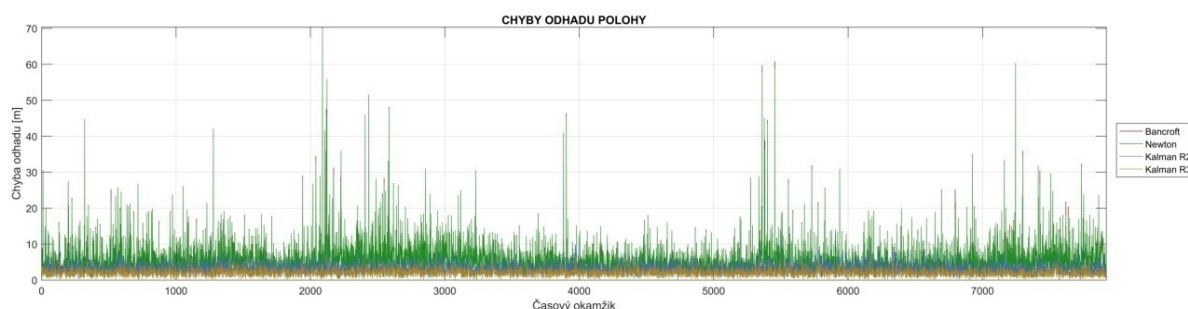
Nutno opět zmínit, že tyto hodnoty jsou totožné po všechny dny vzhledem k statické poloze přijímače a stejným pozicím družic v daný časový okamžik pro jakýkoli siderický den. Z grafů je patrné, že se velmi často hodnota PDOP pohybuje v rozmezí kolem 1,5 až 3. Četnost PDOP klesá postupně od hodnoty 2 až do 6. Nad tuto hodnotu je četnost mizivá. V případě většího stínění by ovšem vyšší hodnoty PDOP byly častější. Graf pod histogramem na stejném obrázku zobrazuje hodnoty PDOP v závislosti na časovém okamžiku. Opět je z něho patrné, že nejčastější hodnota se pohybuje kolem hodnoty 2. Tento graf poslouží ke zjištění závislosti analyzovaných metod pro výpočet odhadu polohy na hodnotě PDOP. Bude tedy možné posoudit, zdali je zhoršení přesnosti dané metody důsledkem (mimo jiné) nevhodného rozmístění družic, či nikoli. Z grafu je zřejmé, že v rozmezí 2 000 – 3 500, dále kolem okamžiků 1 800 a 5 500 a za okamžikem 7 000 jsou hodnoty PDOP relativně vysoké.

Následující obrázek zobrazuje části drah šesti náhodně vybraných družic kolem zjednodušeného modelu Země, tedy kolem rotačního elipsoidu. Na tomto elipsoidu je poloha GPS přijímače reprezentována červeným bodem. Vše je zobrazeno v kartézském souřadnicovém systému v referenční soustavě WGS 84. Tmavě červená barva představuje dráhu družice s označením 29, do které je v simulacích injektována porucha.



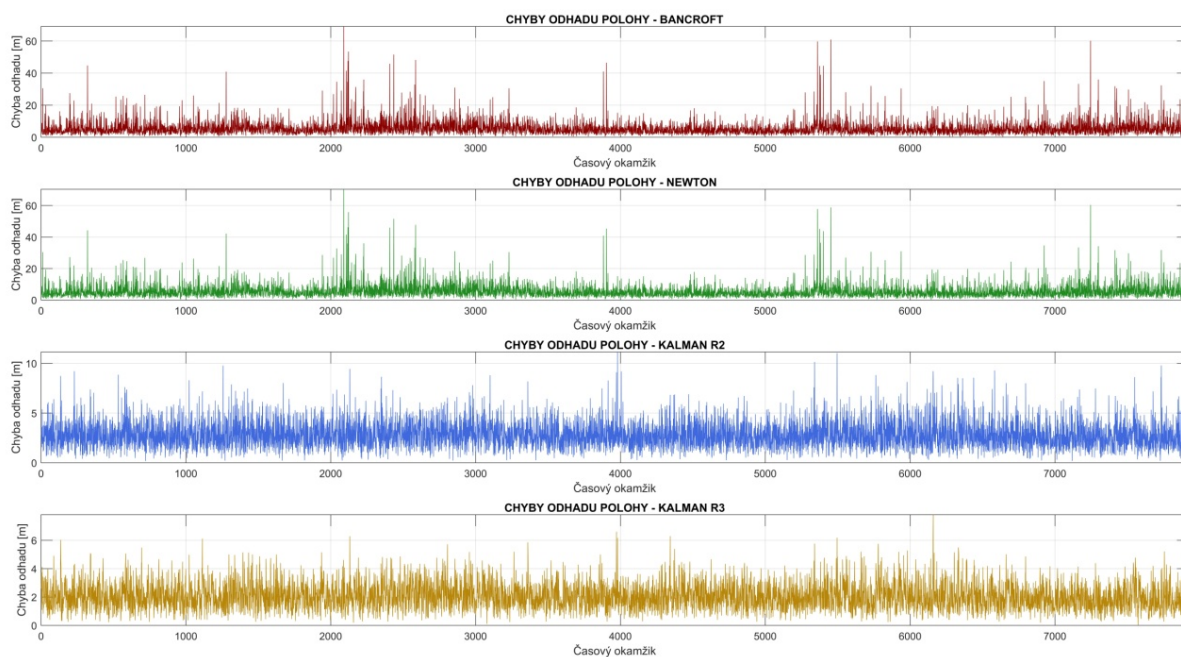
Obr. 23: WGS 84 a dráhy družic

Nejdříve byla provedena analýza přesnosti daných metod bez injektované poruchy v datech při relativně malé hodnotě směrodatné odchylky šumu (simulace1). Obrázek 24 zachycuje časový průběh chyby odhadu polohy pro všechny dané algoritmy v rámci jednoho dne.



Obr. 24: Chyby odhadu polohy (simulace1)

Na dalším obrázku jsou také zachyceny chyby odhadu polohy v rámci dne, tentokrát však pro jednotlivé metody separátně kvůli lepšímu vizuálnímu zobrazení chyb jednotlivých metod. Z obou těchto grafů lze bezpochyby odvodit, že nejmenší chybu odhadu polohy poskytuje rozšířený Kalmanův filtr. Na základě těchto grafů lze rovněž konstatovat, že chyby odhadů vypočtených dle Newtonovy a Bancroftovy metody jsou téměř shodné.

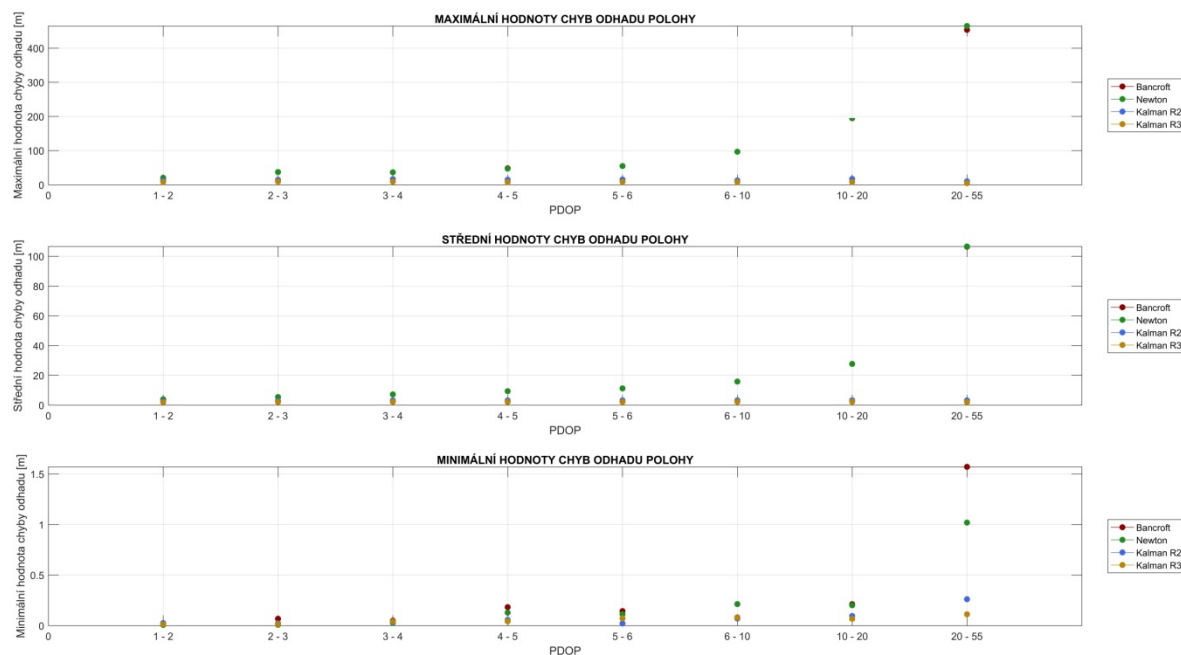


Obr. 25: Chyby odhadu polohy zobrazené separátně (simulace1)

V předchozím textu je uvedeno, že hodnoty PDOP jsou poměrně vysoké v rozmezí okamžiků 2 000 – 3 500, kolem okamžiků 1 800, 5 500 a v rozmezí 7 000 – 8 000, přičemž právě

v těchto okamžicích jsou chyby odhadů stanovených dle Newtonovy a Bancroftovy metody mnohem větší oproti ostatním okamžikům.

Naopak odhad polohy stanovený podle Kalmanova filtru není geometrickým rozmístěním družic příliš ovlivněn, což je zřejmé i z grafu na obrázku 26, jenž znázorňuje maximální, střední a minimální hodnotu chyby klasifikované dle PDOP.

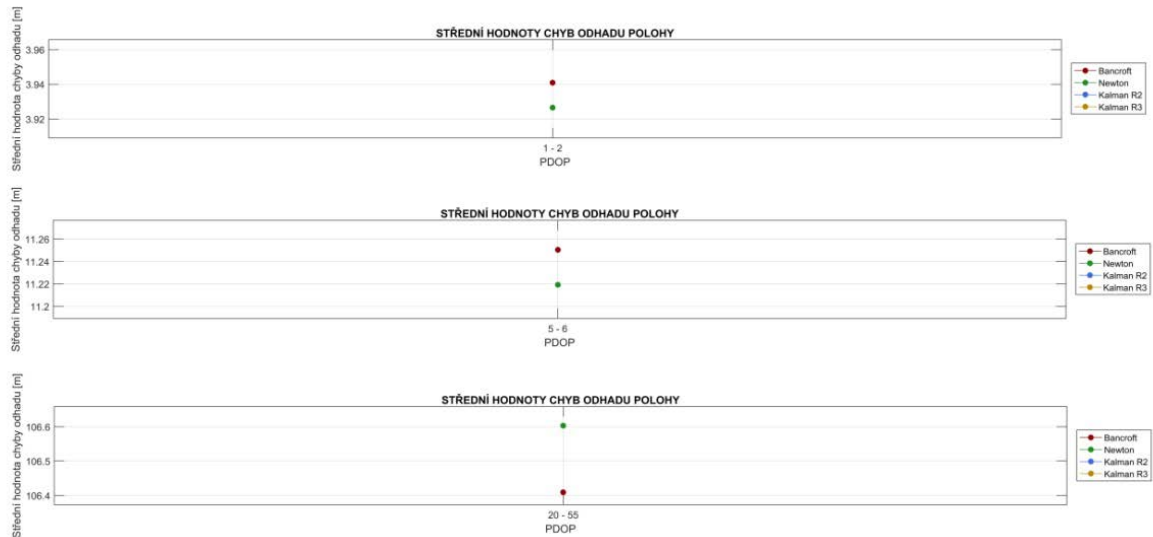


Obr. 26: Chyby odhadu polohy podle PDOP (simulace1)

Z předchozích grafů tedy plyne, že se zvyšující se hodnotou PDOP velmi výrazně narůstá chyba odhadu stanovené podle Bancroftovy a Newtonovy metody, na rozdíl od Kalmanova filtru. Na následujícím obrázku č. 27 je zobrazen detail výše uvedeného grafu, který je zaměřen na Bancroftovu a Newtonovu metodu pro nízké a vysoké hodnoty PDOP. Z těchto grafů je zřejmé, že pro nižší hodnoty PDOP (v rozmezí 1 až 2 či 5 až 6) je střední hodnota chyby odhadu menší při výpočtu Newtonovou metodou. Nicméně Bancroftův algoritmus poskytuje přesnější výsledky pro vyšší hodnoty PDOP (kolem 20 až 55) než Newtonova metoda.

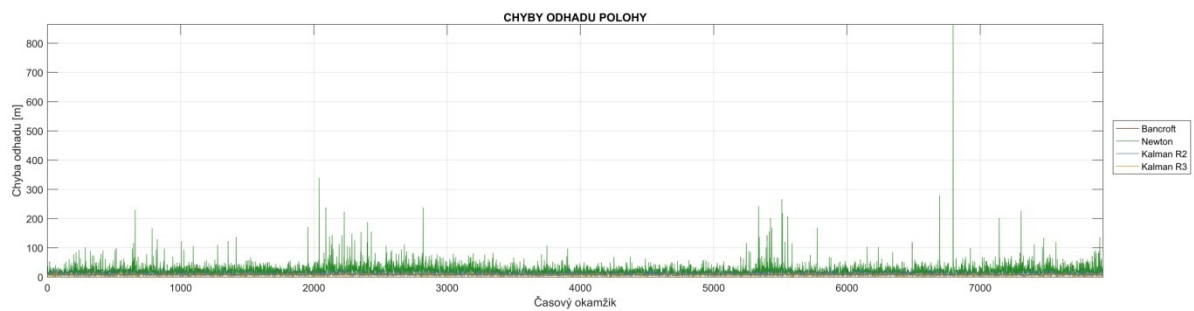
Nutno dodat, že rozdíl chyb těchto metod je v řádu desetin metru, čili pokud nejde o implementaci metody pro přesná geodetická měření, není nutné rozmyslet se nad tím, která z těchto metod bude výhodnější. Hypotéza z kapitoly 2.2.1 se tedy týká velmi vysokých hodnot. Lze tedy spíše říci, že se tyto metody chovají totožně.

Obecně v případě relativně malé směrodatné odchylky šumu bez poruch v naměřených datech je nejvýhodnější implementovat rozšířený Kalmanův filtr (nejlépe pro model 3. řádu), který není příliš ovlivněn geometrickým rozmístěním družic tak jako zbylé dvě metody.



Obr. 27: Chyby odhadu polohy podle PDOP – detail (simulace1)

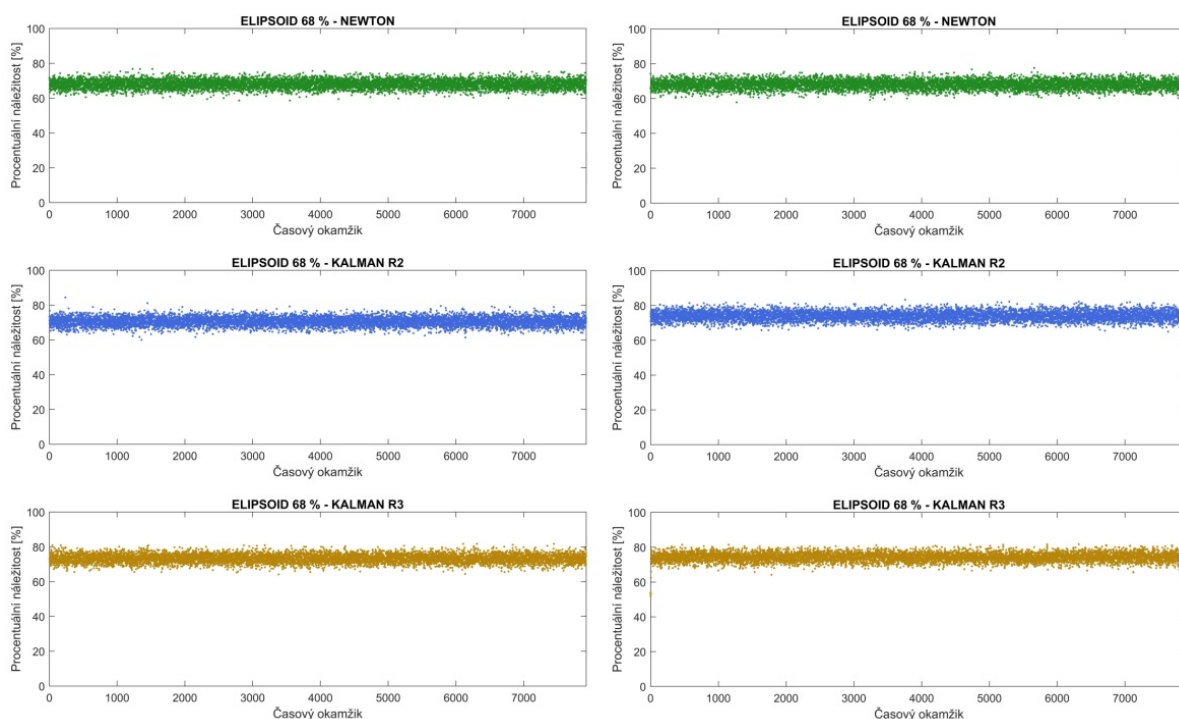
Následující simulace byla provedena opět bez přítomnosti poruch v datech, ovšem s větší směrodatnou odchylkou šumu ke zjištění, jakým způsobem ovlivní parametr šumu výpočet odhadu polohy pro všechny analyzované algoritmy (simulace2). Obrázek 28 zachycuje průběh chyby odhadu polohy pro tuto simulaci v rámci jednoho dne.



Obr. 28: Chyby odhadu polohy (simulace2)

Je očekávané, že větší hodnota STD šumu výrazně ovlivní přesnost odhadu polohy, ovšem při porovnání chyb odhadu polohy lze dojít ke stejnému závěru jako při předchozí simulaci, tedy že nejvíce přesným algoritmem je rozšířený Kalmanův filtr, a to konkrétně pro model 3. řádu. Na jedné z následujících stránek jsou zobrazeny grafy pro procentuální příslušnost vůči elipsoidu 68 %, aby bylo možné posoudit, zdali jsou odhady správné.

Jednou z hypotéz uvedených na začátku této kapitoly bylo, že hodnota směrodatné odchylky šumu bude mít sice vliv na přesnost odhadu polohy, ovšem neměla by příliš ovlivnit procentuální náležitost vůči kovariančnímu elipsoidu. Tato náležitost je prezentována na grafu na níže uvedeném obrázku, kde je v levé části zobrazena procentuální náležitost z předchozí simulace (pro STD šumu o hodnotě 2,5 m) a v pravé části z této simulace (pro STD šumu 10 m). Jak lze spatřit, i tato hypotéza byla potvrzena.

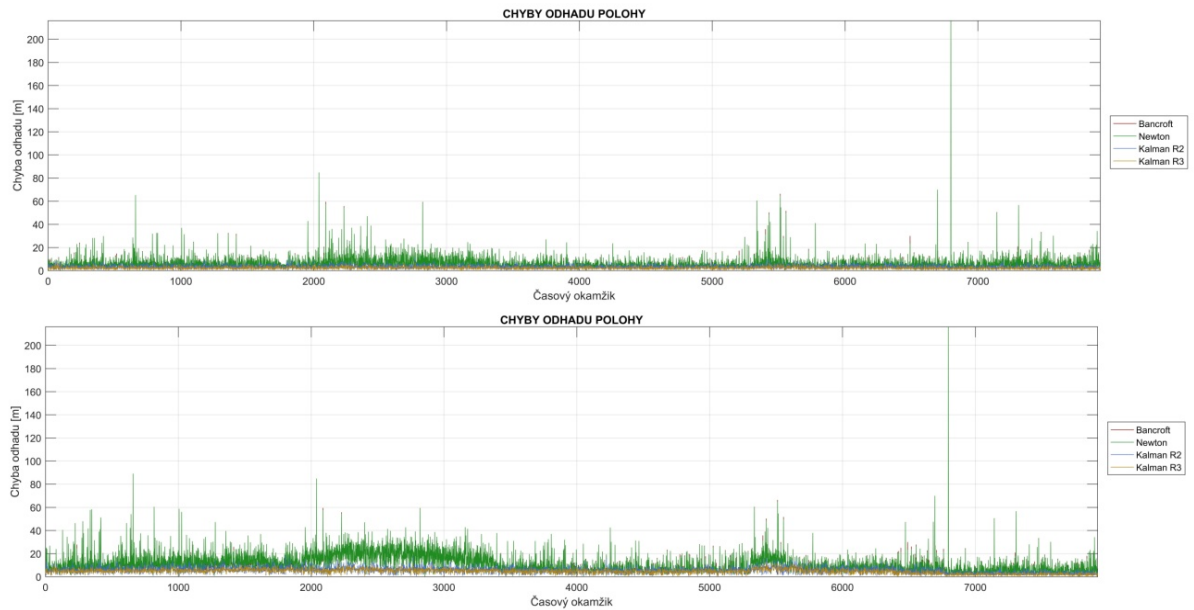


Obr. 29: Procentuální náležitost vůči kovariančnímu elipsoidu (vlevo sim.1, vpravo sim.2)

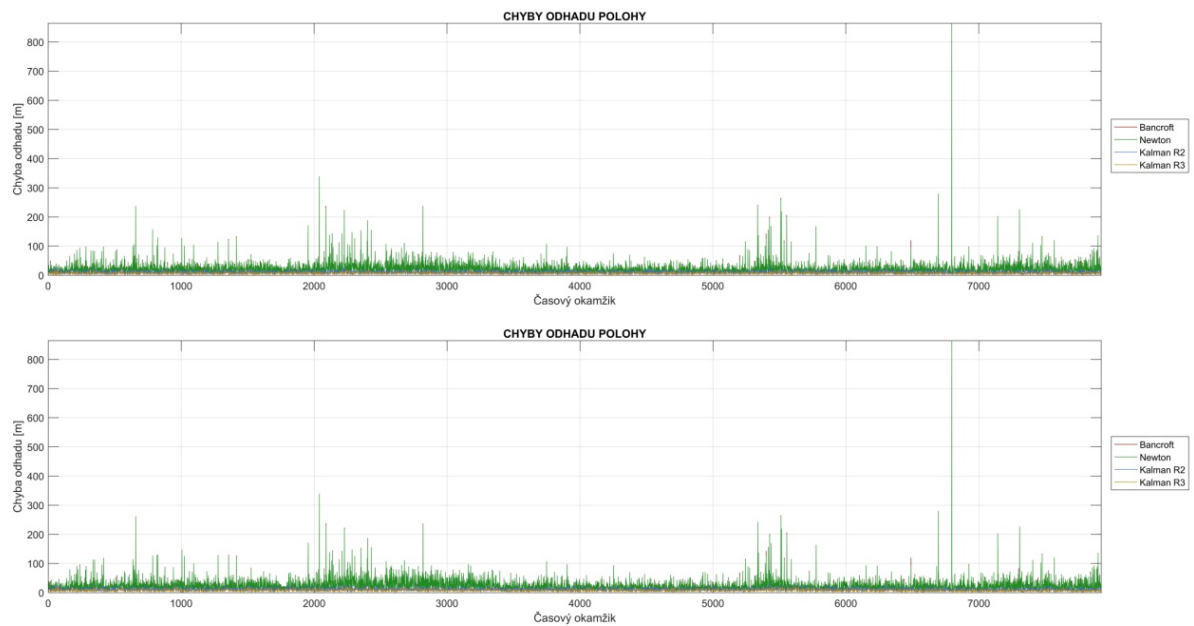
Následující čtyři simulace byly provedeny pro různé kombinace STD šumu a injektované chyby v datech jedné družice s cílem analyzovat reakce daných metod na přítomnost poruchy. Obrázek č. 30 zachycuje průběh simulací se směrodatnou odchylkou šumu 2,5 m při injektované poruše 2,5 m a 10 m. (simulace3 a 4). Obrázek č. 31 zachycuje průběh simulací se stejnými poruchami, ovšem s hodnotou STD šumu 10 m. Z těchto grafů a z následující tabulky č. 2 lze usoudit, že chyba určení odhadu polohy je nejmenší opět při výpočtu pomocí rozšířeného Kalmanova filtru. Porovnáním horního grafu na obrázku 30 (STD 2,5 m s poruchou 2,5 m) s grafem na obrázku 24 (STD 2,5 m bez poruchy), lze konstatovat, že se od sebe prakticky neliší.

Tím je potvrzena další hypotéza z úvodu této kapitoly, který se týká částečného ukrytí chyby v šumu v případě, kdy je hodnota přítomné poruchy shodná s hodnotou směrodatné odchylky šumu. Totéž lze pozorovat i při porovnání dolního grafu na obrázku 31 s grafem na obrázku

28. Podobný průběh chyb odhadu polohy je zachycen na horním grafu na obrázku 31, poně-  
vadž v tomto případě byla porucha menší než směrodatná odchylka šumu. Značný vliv měla  
porucha 10 m v případě směrodatné odchylky šumu 2,5 m (viz dolní graf na obrázku č. 30).  
Zde je vidět, že tato porucha ovlivnila odhad polohy dle Bancroftovy a Newtonovy metody  
v těch okamžicích, kde byla PDOP vysoká. Při analýzách by bylo vhodné ověřit, jaký vliv by  
měla družice 29 v těchto okamžicích. K tomu se používá tzv. deltaDOP, což je kvadratický  
rozdíl DOP pro původní kombinaci družic a DOP pro družice, kde daná družice 29 chybí.



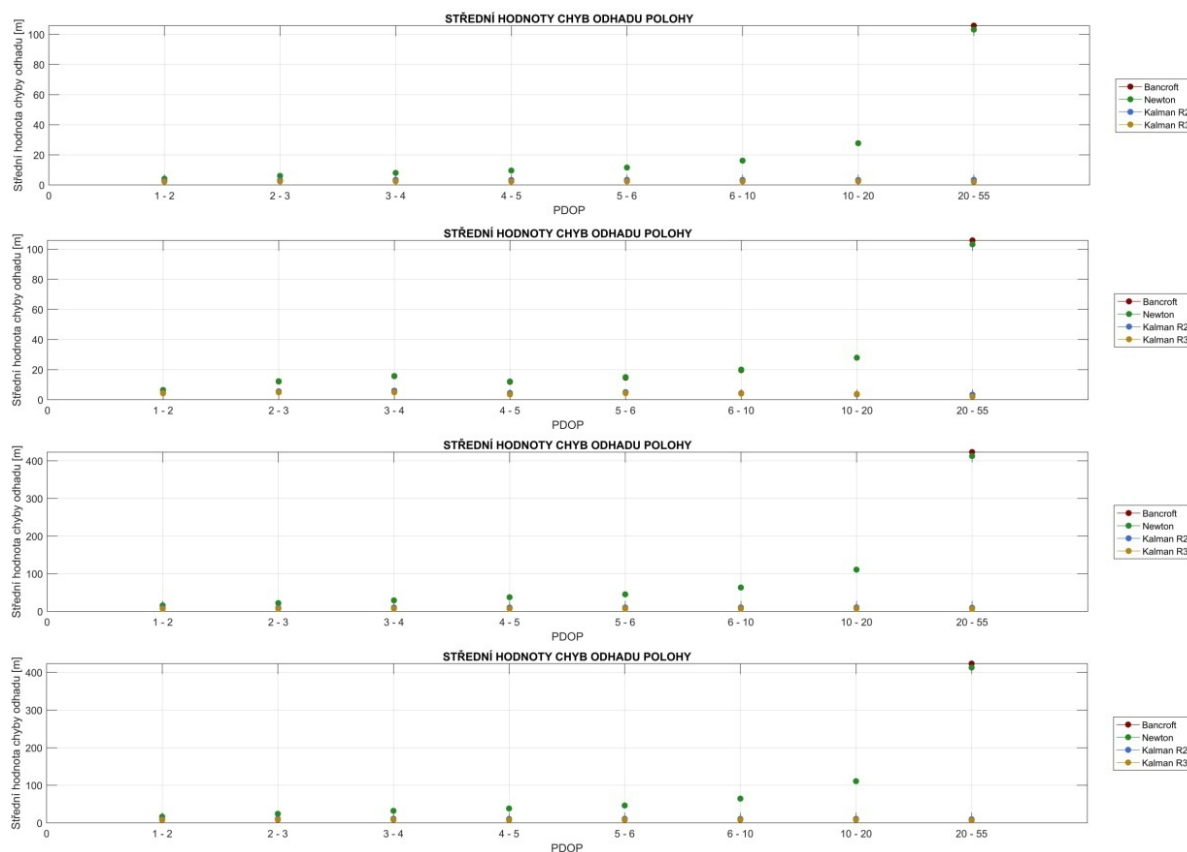
Obr. 30: Chyby odhadu polohy (nahore sim.3, dole sim.4)



Obr. 31: Chyby odhadu polohy (nahore sim.5, dole sim.6)

Zejména kolem okamžiku 5 500, kdy nebylo geometrické uspořádání družic příliš optimální, byly chyby odhadu vypočtené dle Kalmanova filtru během simulace také výrazně vyšší.

Graf na níže uvedeném obrázku zachycuje střední hodnoty chyb odhadu polohy pro dané matematické algoritmy klasifikované dle hodnoty PDOP pro simulace 3 až 6. V příloze jsou poté uvedeny také maximální a minimální hodnoty těchto chyb.



Obr. 32: Střední hodnoty chyb odhadu polohy (shora dolů sim.3 až sim.6)

Z těchto grafů je zřejmý vliv rozmístění družic na Bancroftův a Newtonův algoritmus, kdy střední hodnota chyb odhadů vypočtených dle těchto dvou metod roste se vzrůstající hodnotou PDOP. Chyby odhadů dle Bancroftovy metody nejsou v grafu viditelné, poněvadž jsou překryté chybami odhadu dle Newtonovy metody. Liší se v řádech pouhých desetín metru, jsou tedy v podstatě shodné.

Velmi zajímavé jsou ovšem střední hodnoty chyb odhadů dle Kalmanova filtru, kdy pro oba řády modelu platí, že chyba odhadu polohy není hodnotou PDOP příliš ovlivněna. V tabulce č. 2 jsou přehledně vypsány maximální, střední a minimální hodnoty chyb odhadů pro všechny analyzované metody dle PDOP. Kalmanův filtr pracuje s pamětí, tudíž v následujících okamžicích, kdy je hodnota PDOP vyšší již využívá předchozí, relativně přesný odhad. Tím

pádem jsou maximální hodnoty chyb odhadů pro vysoké hodnoty PDOP nižší než u ostatních metod. Hodnoty jsou zaokrouhleny na jedno desetinné místo, pouze v případě minimální hodnoty chyby odhadu došlo k zaokrouhlení na dvě desetinná místa.

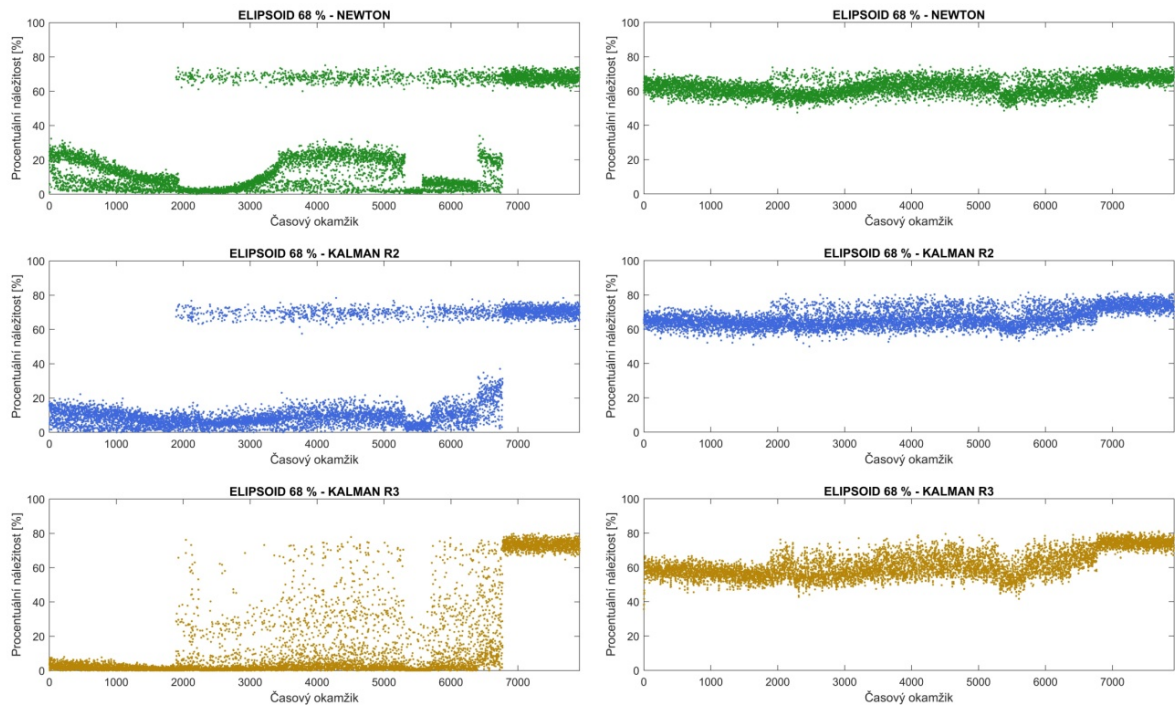
Tab. 2: Hodnoty chyb odhadu polohy (simulace4)

	PDOP	Bancroft [m]	Newton [m]	Kalman R2 [m]	Kalman R3 [m]
1 až 2	MIN	0,05	0,05	0,04	0,02
	MEAN	6,4	6,4	4,5	4,2
	MAX	27,5	28,2	18,4	13,8
2 až 3	MIN	0,04	0,04	0,04	0,03
	MEAN	12,1	12,2	5,5	4,9
	MAX	49,3	49,8	25,1	18,9
3 až 4	MIN	0,07	0,07	0,04	0,03
	MEAN	15,6	15,7	5,9	4,9
	MAX	58,4	58,2	24,3	15,5
4 až 5	MIN	0,18	0,12	0,11	0,09
	MEAN	11,8	12,1	4,4	3,5
	MAX	65,5	68,2	20,2	13,6
5 až 6	MIN	0,08	0,35	0,09	0,07
	MEAN	14,6	14,9	4,9	4,3
	MAX	79,4	79	21,2	14,3
6 až 10	MIN	0,13	0,18	0,09	0,08
	MEAN	19,6	19,9	4,3	4
	MAX	111,3	111,2	18,9	13,2
10 až 20	MIN	0,15	0,45	0,12	0,11
	MEAN	27,8	27,9	3,5	3,7
	MAX	162,5	162,9	19,2	10,9
20 až 55	MIN	1,28	0,77	0,36	0,11
	MEAN	105,8	103,1	3,38	2
	MAX	436,7	416,6	11,9	5,4

Graf na následujícím obrázku zachycuje procentuální náležitost vůči kovariančnímu elipsoidu 68 % pro simulace4 (STD 2,5 m s poruchou 10 m) a pro simulace6 (STD 10 m s poruchou 10 m). Na tomto grafu je vidět rozdíl náležitosti při simulaci se stejně velkou injektovanou poruchou, avšak rozdílnou hodnotou STD šumu.

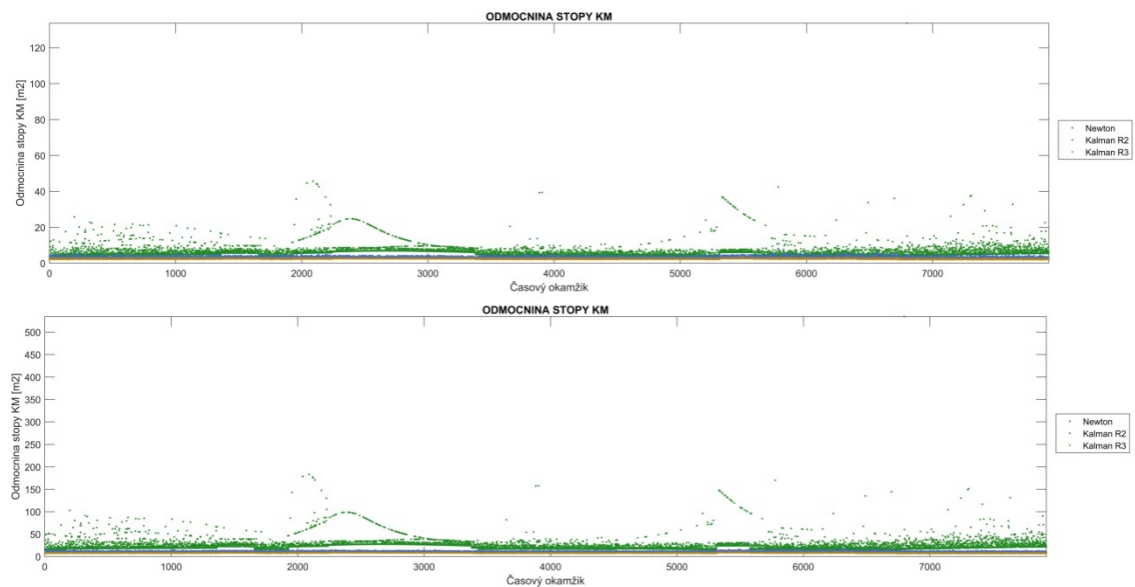
Pokud je totiž hodnota chyby podobná směrodatné odchylce šumu, nebude to na správnost odhadu (náležitost vůči kovariančnímu elipsoidu) mít takový vliv jako při poruše vyšší než STD šumu, a to právě z toho důvodu, že se chyba částečně ukryje v šumu, jak již ostatně bylo řečeno v předešlém textu. Zároveň je z grafu patrné, že od okamžiku kolem 6 900 již není poruchová družice dostupná.





Obr. 33: Procentuální náležitost vůči kovariančnímu elipsoidu (vlevo sim.4, vpravo sim.6)

Graf na níže uvedeném obrázku zachycuje odmocninu stopy kovarianční matice pro Newtonovu metodu a rozšířený Kalmanův filtr pro oba řády modelu. Stopa byla spočtena z důvodu snazšího porovnání KM. Odmocnina stopy je zachycena pro simulace 4 a 5.

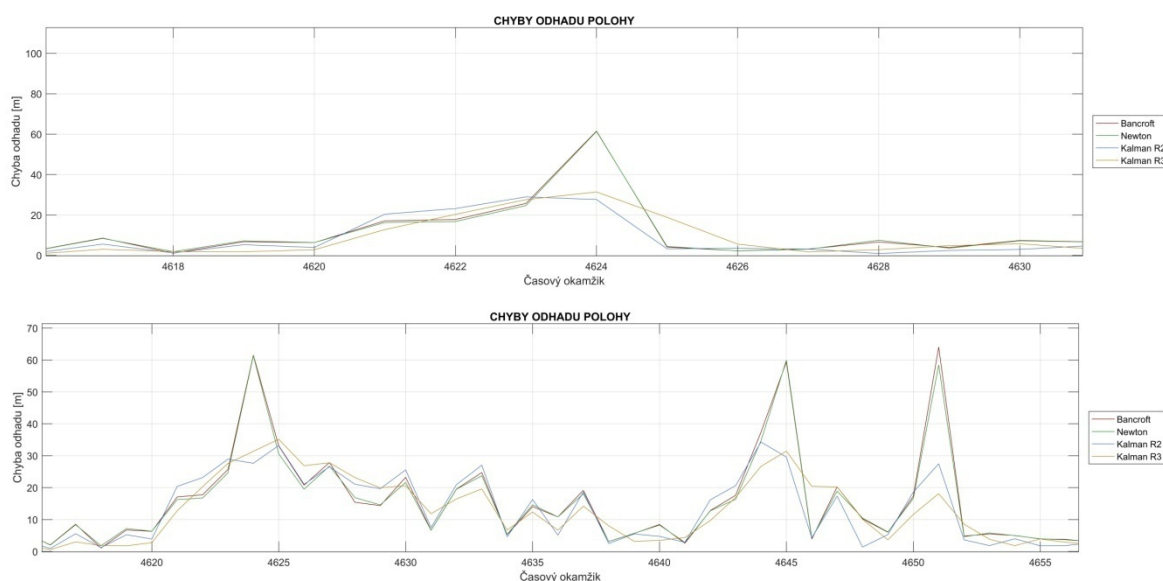


Obr. 34: Odmocnina stopy KM (nahore sim.4, dole sim.5)

Z obou grafů je patrný vliv rozmístění družic na celkovou přesnost odhadu polohy. V případě většího šumu je pochopitelně snížena přesnost, avšak i přes to jsou hodnoty rozptylu odhadů

dle Kalmanova filtru v úzkém rozmezí, naproti od Newtonovy metody, u které jsou rozptýly odhadů v průběhu celé simulace poměrně značně „rozeseté“.

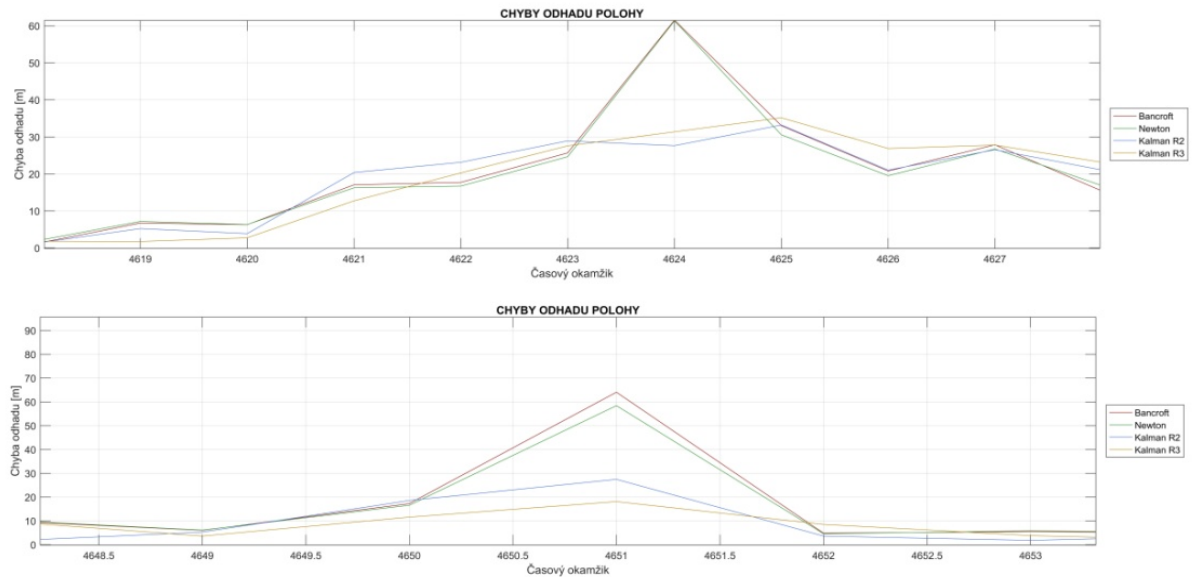
Poslední část simulací byla provedena za účelem posouzení reakce metod na kratší a delší působení extrémní poruchy. Cílem bylo potvrdit, případně vyvrátit hypotézu týkající se paměti Kalmanova filtru. Konkrétně zdali skutečně bude Kalmanův filtr na chybu reagovat s určitým zpožděním a zdali bude s chybou pracovat i jistou chvíli po jejím odeznění. Graf na obrázku 35 zobrazuje výšek několika časových okamžiků pro chyby odhadu polohy vypočtené dle daných metod.



Obr. 35: Chyby odhadu polohy (nahore sim.7, dole sim.8)

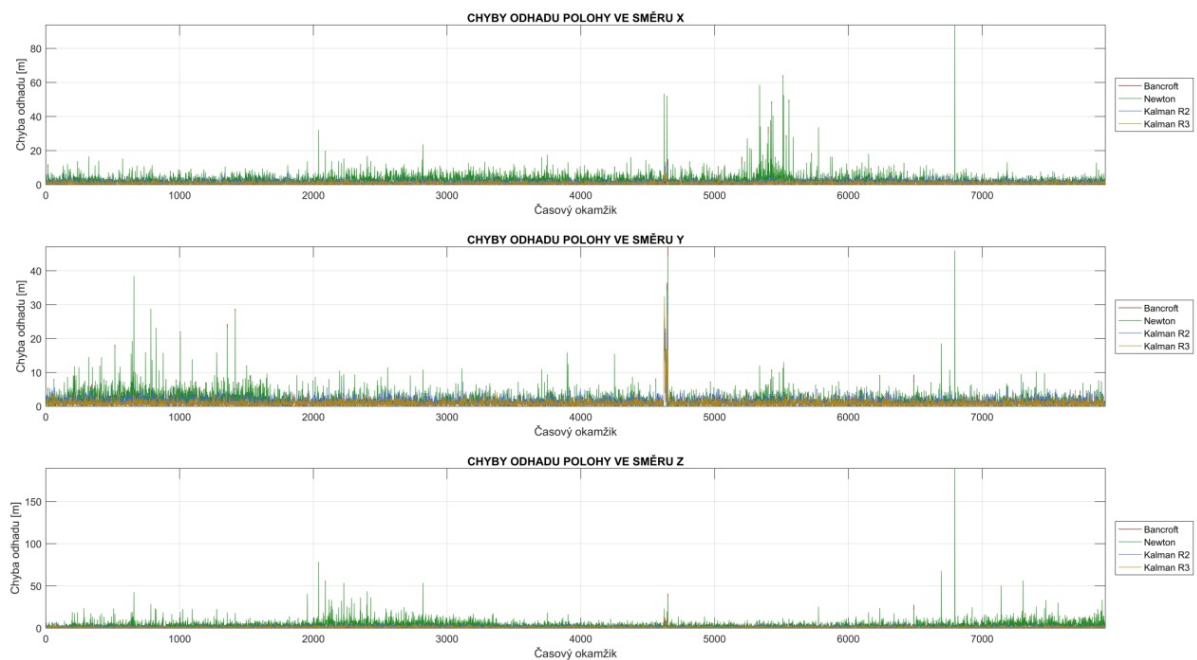
Při bližším pohledu je zřetelně vidět rozdíl mezi Kalmanovým filtrem a zbylými dvěma metodami. Extrémní chyba 50 m byla injektovaná v časový okamžik 4 620, přičemž tato porucha se výrazně projevila ihned, kdy se chyba odhadu stanoveného dle Newtonovy a Bancroftovy metody výrazně zvýšila na rozdíl od Kalmanova filtru (zejména u modelu druhého řádu). Zde došlo ke zvýšení chyby s určitým zpožděním, a to právě z důvodu toho, že pracuje s pamětí, čili využívá předchozí odhad a aktuální naměřená data ke korekci aktuálního odhadu polohy. Totéž platí i po odeznění poruchy v okamžiku 4 650 v simulace8. Zatímco chyba Newtonovy a Bancroftovy metody se sníží okamžitě v následující časový okamžik, chyba odhadu dle Kalmanova algoritmu (zejména u modelu třetího řádu) je snižována s určitou prodlevou, a to opět z důvodu práce s pamětí.

Obrázek 36 zachycuje detail grafu pro chyby odhadu polohy pro časové okamžiky kolem začátku (horní graf) a konce (dolní graf) působení poruchy v simulace8. Tímto byla potvrzena další hypotéza z úvodu třetí kapitoly. Je ale nutné si uvědomit, že se v průběhu okamžiků může měnit konstelace, což také může vyvolat prudší skok v chybě odhadu polohy.



Obr. 36: Chyby odhadu polohy – detail (simulace8)

Níže jsou také zobrazeny dva grafy týkající se průběhu chyby odhadu polohy v jednotlivých směrech X, Y a Z pro simulace8.



Obr. 37: Chyby odhadu polohy – detail (simulace8)

Z tohoto grafu je zřetelné, že odhad polohy dle Newtonovy a Bancroftovy metody je poruchou ovlivněn ve všech směrech zhruba stejně, avšak u odhadu dle Kalmanova filtru je odhad polohy ve směru Y ovlivněn poruchou mnohem více než ve zbylých dvou směrech. To může být způsobem např. tím, jak se zeměpisná šířka, délka a výška rozkládá do souřadnic X, Y a Z v oblasti střední Evropy.

Souhrnně lze tedy konstatovat, že výstupy (tabulky a grafy) provedených simulací s vygenerovanými daty potvrdily všechny hypotézy uvedené v úvodu třetí kapitoly včetně hypotézy z kapitoly 2.2.1.

Bancroftova algebraická metoda podává velmi podobné výsledky jako Newtonova iterační metoda. Přestože Bancroftova metoda skutečně poskytuje přesnější odhad polohy v případě zhoršeného geometrického rozmístění družic, je rozdíl chyb těchto odhadů v řádech desetiný jednoho metru, tudíž s ohledem na lokalizaci vozidel na železniční síti lze říci, že přesnost obou metod je ve své podstatě stejná.

Ve všech provedených simulacích poskytoval Kalmanův filtr odhady polohy s nejmenšími chybami a rozptylem. Ideální je použití Kalmanova filtru pro třetí řád modelu. Zároveň byl prokázán nepříliš významný vliv geometrického uspořádání družic na přesnost Kalmanovy filtrace. Simulace také dokázaly opožděnou reakci Kalmanova filtru na přítomnost poruchy a rovněž jisté zpoždění této metody po odeznění poruchy.

Dále byly posouzeny vlivy různých kombinací STD šumu a injektované poruchy na přesnost odhadu polohy dle daných algoritmů. Bylo prokázáno, že hodnota STD šumu sice ovlivní přesnost odhadu, nicméně v případě, kdy nepřítomnosti poruchy není ovlivněna procentuální náležitost vůči kovariančnímu elipsoidu.

Nakonec bylo stanoveno, že odhad polohy vypočtený pomocí Bancroftovy a Newtonovy metody se pohybuje kolem referenční polohy přijímače nestranně, na rozdíl od Kalmanova filtru, u kterého je odhad polohy ve směru Y ovlivněn mnohem více než ve směrech X a Z, což může souviset s tím, že GPS je v odhadu nadmořské výšky méně přesná než u odhadu zeměpisné šířky a délky, přičemž v oblasti střední Evropy se tato výška projevuje spíše v X a Z než v Y.

## 4 Ověření metod určení odhadu polohy

Při testování algoritmů pro výpočet odhadu polohy GNSS přijímače mají simulace mnoho předností, kterými disponují oproti reálným experimentům. Jsou výhodné zejména díky možnosti zopakování provedení měření za stejných podmínek, dále díky variabilitě scénářů, které lze nastavit, jako je lokace, okolní faktory, poruchy apod. Simulace bývají také obvykle časově a finančně méně náročné oproti měřením v praxi, kdy je nutné obstarat zejména testovací plochu či dráhu, vozidlo, příslušný personál a další vybavení.

I přes tyto výhody jsou simulace stále pouze napodobením skutečného stavu či procesu. Je tedy vždy vhodné a nutné provést také reálné měření v terénu, a to pro ověření, zdali dané algoritmy skutečně poskytují požadované výstupy přímo v reálu. V takovém případě jde o ověřovací provoz. V případě vývoje GPS přijímače určeného pro železniční dopravu je samozřejmě nejen nutné jej otestovat na železniční síti, nejlépe přímo na konkrétní trati.

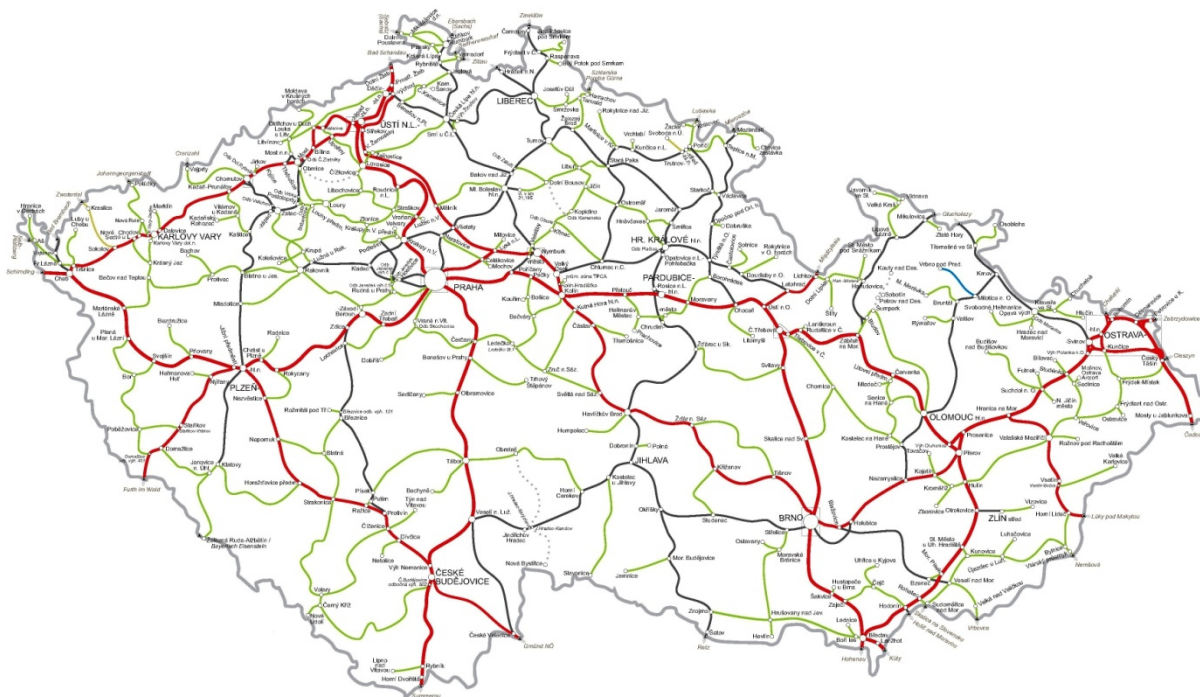
Ještě před provedením reálného experimentu je nutné si definovat předpoklady pro měření, předmět zkoumání a vyhodnocování a v neposlední řadě také cíle experimentu. Tedy co se očekává a co by mělo být výstupem měření (např. hodnota požadované přesnosti odhadu polohy, nebo reakce testovaných metod na různé okolní faktory). Výsledky experimentu jsou poté porovnány s očekávanými hodnotami a výstupy poté, stejně jako v simulacích, potvrdí, nebo vyvrátí dané hypotézy.

Vzhledem k tomu, že tato práce byla iniciativou společnosti AŽD Praha s.r.o., je následující text věnován železniční dopravě, a to konkrétně popisu možnosti využití GNSS na železničních tratích, návrhu a popisu reálného experimentu pro ověření daných metod určení odhadu polohy. Nakonec budou vyhodnoceny výstupy z reálného měření v terénu, které budou porovnány s výstupy již provedených simulací. Ještě před samotným návrhem postupu reálného experimentu, je vhodné se zmínit obecně o železniční dopravě a o lokalizaci vozidel na železniční síti, a to s důrazem na možnosti využití GNSS.

### 4.1 Lokalizace vozidel na železničních tratích

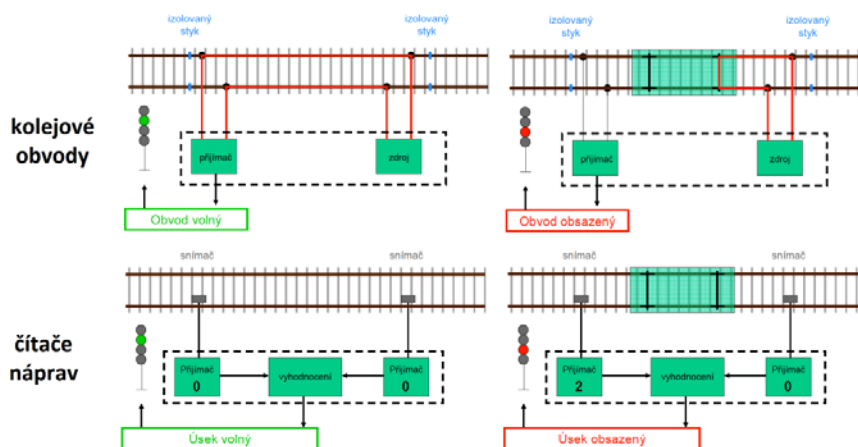
Celková délka železničních tratí v České republice odpovídá necelým 9 500 km, přičemž 38 % z toho jsou tratě regionální (tedy o délce kolem 3 600 km). Regionální tratě jsou méně vytížené, než tratě zařazené do sítě TEN-T. Vzhledem k tomu, že na těchto tratích není značný provoz, funguje zde většinou způsob zabezpečení jízdy vlaku mezi stanicemi založený

na telefonickém dorozumívání (příhláška a odhláška). Následující obrázek zachycuje kategorie drah na území ČR, přičemž červenou barvou jsou vyznačeny celostátní tratě zařazené do systému TEN-T, šedou barvou jsou reprezentovány ostatní celostátní dráhy. Regionální tratě jsou vyznačeny zeleně, modře a žlutě, přičemž barvy odlišují provozovatele dráhy. [52]



Obr. 38: Kategorie drah na území ČR [53, upraveno]

Pro řízení a zabezpečení železničního provozu je základní vstupní informací znalost aktuální polohy drážního vozidla. K získání této informace o poloze jsou využívány technické prostředky (kolejové obvody nebo počítače náprav – viz obr. 39). Avšak na regionálních tratích je aktuální poloha zjišťována stále ještě příslušným personálem.



Obr. 39: Kolejové obvody a počítače náprav [54, upraveno]

V minulosti došlo na regionálních tratích k několika závažným nehodám, při kterém se srazila drážní vozidla. Například na trati Čičenice – Volary v jižních Čechách došlo během posledních 12 let ke třem nehodám, kdy se ve dvou případech srazily osobní vlaky. Ve třetím případě došlo ke srážce osobního vlaku s vlakem manipulačním. Všechny tyto nehody byly důsledkem neoprávněného odjezdu vlaku ze stanice do obsazeného traťového oddílu. Došlo tedy k selhání lidského činitele ať už na straně dispečera nebo strojvedoucího. [55]

#### 4.1.1 Zabezpečovací zařízení v železniční dopravě

Lokalizaci drážních vozidel lze rozdělit do dvou oblastí zájmu, a to pro systémy informační a systémy zabezpečovací. Ve druhém případě je kladen značný důraz na bezpečnost a spolehlivost celého systému. V železniční dopravě je bezpečnost dána pěti úrovněmi integrity bezpečnosti (SIL 0 až SIL 4) dle normy ČSN EN 50129. Jednoduše řečeno úroveň SIL kvantifikuje omezení rizika. Pro zabezpečovací zařízení na železnici je vyžadována nejvyšší (pátá) úroveň, čili nesmí se vyskytnout více než jedna nebezpečná chyba za 10 000 let. [56, 57]

Železniční infrastruktura byla v každé zemi budována a rozvíjena separátně. To znamená, že v současné době nejsou signalizační a zabezpečovací systémy v rámci Evropy kompatibilní. Pro zvýšení interoperability, a tím konkurenceschopnosti železniční dopravy vznikla iniciativa ERTMS, jejíž součástí je GSM-R (globální systém pro mobilní komunikaci na železnici) a ETCS (systém evropského vlakového zabezpečovacího zařízení). ETCS je považován z hlediska TSI za celoevropský standard, který by měl do budoucna nahradit více než 20 různých národních vlakových zabezpečovacích zařízení a umožnit tak jízdu vlaku po Evropě bez nutnosti výměny hnacího vozidla. S tímto systémem se ale počítá spíše na tranzitních koridorech vzhledem k tomu, že jde o poměrně finančně nákladné řešení. ETCS se skládá z mobilní části ve vozidle a z infrastrukturní části umístěné v kolejišti. Konkrétně jde o fyzické prvky zvané balízy, jež slouží jako referenční body a také k přenosu informací na vozidlo. [58]

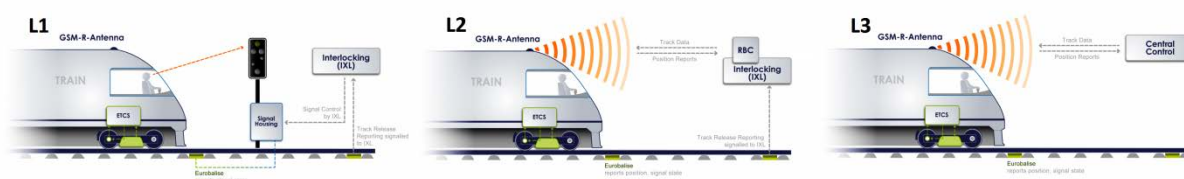


Obr. 40: Infrastrukturní a mobilní část systému ETCS [59, 60]

V systému ETCS jsou definovány tři základní úrovně, jež jsou zachyceny na obrázku č. 41. První úroveň (ETCS L1) slouží pouze jako doplněk k současnému národnímu zabezpečovacímu zařízení. Přepínatelné balízy slouží k bodovému předávání informací (návěsti, traťové poměry, povolení k jízdě) na vozidlo a rovněž korigují polohu vlaku. Balízy jsou k zabezpečovacímu zařízení propojeny kabely. Celistvost vlaku je v tomto případě zjišťována klasickým způsobem pomocí kolejových obvodů, případně počítačů náprav. Tato úroveň také vyžaduje světelná návěstidla. Palubní část ETCS měří aktuální rychlost jízdy, zjišťuje veličiny pro stanovení aktuální polohy vozidla a přijímá informace od infrastrukturní části. Dojde-li k překročení hranice oblasti povolení k jízdě nebo překročení nejvyšší dovolené rychlosti, systém varuje strojvedoucího a případně provede zásah do řízení vozidla. [58]

U druhé úrovně (ETCS L2) již nedochází k bodovému, ale ke kontinuálnímu obousměrnému přenosu informací přes GSM-R. Balízy v tomto případě pouze korigují polohu vozidla. Sice již nejsou nutná světelná návěstidla, nicméně kontrola celistvosti vlaku probíhá stále konvenčními prostředky. Tato úroveň je zatím testována na některých úsecích tranzitních železničních koridorů v ČR. [58]

Nejvyšší úroveň (ETCS L3) rovněž nevyužívá světelná návěstidla a kontrola integrity vlaku probíhá přímo ve vozidle v palubním počítači. Nejsou tedy nutné kolejové obvody na trati. Centrála zaznamenává polohu vlaku a uděluje povolení k jízdě. Přenos informací je opět obousměrný pomocí GSM-R. Značným přínosem této úrovně je zvýšení propustnosti tratí, a to díky jízdě v tzv. pohyblivých traťových oddílech, díky čemuž lze optimalizovat rozestupy mezi vlaky. Toto řešení je však stále předmětem výzkumu. [58]



Obr. 41: Úrovně systému ETCS [61]

Vlakový zabezpečovací systém ETCS není ovšem efektivní pro regionální tratě zejména kvůli investičním a provozním nákladům. Na těchto tratích je tedy cílem redukovat, případně eliminovat vliv lidského činitele, ovšem s minimální finanční náročností. Zde se právě nabízí možnost využití družicové navigace. Do budoucna se také u ETCS uvažuje o doplnění družicové navigace v podobně flexibilních virtuálních balíz, jejichž poloha bude uložena v palubním



počítači. Pokud drážní vozidlo projede danou virtuální oblastí, bude systému poskytnuta stejná informace, jako kdyby projelo nad pevnou balízou.

#### 4.1.2 Využití GNSS pro lokalizaci drážních vozidel

Družicová navigace v železničních aplikacích tedy představuje nový směr s cílem nahradit finančně nákladnou infrastrukturu podél trati palubními systémy, dále vytvořit předpoklady pro interoperabilitu, poskytnout účinné nástroje pro řízení provozu a zvýšit bezpečnost.

Existuje mnoho služeb, které mohou využívat systém GNSS. Jde například o informační služby pro cestující (údaj o poloze vlaku na obrazovkách ve stanicích), dále pak odbavovací systémy (tarifní lokace), diagnostické systémy, měření spotřeby jízdy, asistenční systémy strojvedoucího, případně i podpůrné služby pro automatické vedení vlaku.

Bohužel zatím není poskytnuta garance parametrů systému GNSS, které jsou důležité pro bezpečnost na železnici. Systém GNSS totiž nemusí zatím splňovat podmínky vztahující se k normě ČSN EN 50129. Proto tyto systémy zatím nemohou být použity pro zabezpečení jízdy vlaků. Mohou však sloužit jako doplňkový nástroj, případně mohou být využity pro informační systémy. [62]

Ve stanicích s více dopravními a manipulačními kolejemi a na více kolejných tratích, u kterých bývá osová vzdálenost kolejí standardně 4 m, je nezbytné znát polohu s relativně vysokou přesností (kolem 1 metru), aby bylo zřetelné, na které koleji se drážní vozidlo nachází. Ovšem takováto přesnost může být pro GNSS problém.

#### Příklady využití GNSS v praxi

Od roku 2000 jsou řešeny desítky evropských projektů na téma využití GNSS na železnici (projekt APOLO, STARS, GRAIL, a další). V Rakousku je již využíván systém na bázi GNSS pro podporu dispečerského řízení na jednokolejných tratích. Systém je koncipován jako automatizovaná podpora pro kontrolu oprávnění k jízdě. Pokud se kolejová vozidla vyskytnou na trati v limitní vzdálenosti ke konci povolení, spustí se ve vozidle alarm. Limitní vzdálenost je počítána na základě aktuální rychlosti vozidel, z čehož je odvozena potřebná zábrzdňá vzdálenost. Ve vlaku je samostatné zařízení zjišťující polohu dle GPS a odometru.

V České republice je vyvíjen společností AŽD Praha s.r.o. systém na obdobném principu. Systém nese název Radioblok RB1 a je v současné době testován na trati Číčenice – Volary, která byla vybrána kvůli závažným nehodám, jež se v minulosti staly (viz předešlý text). Tato trať o celkové délce 56 km je jednokolejná. Na této trati je již několik let implementován

Radioblok verze RB0+, který sice vnitřně využívá GNSS, nikoli však pro bezpečnou informaci, nicméně i tak pomáhá ke zvýšení bezpečnosti. Systém Radiobloku se skládá z mobilní části (RBV), jež je součástí vozidla a stacionární části (RBS) umístěné na dispečerském stanovišti ve stanici Prachatice. Do RBS jsou zasílány z hnacího vozidla zprávy s informací o poloze vlaku na trati a z radioblokové centrály dostává vedoucí drážní vozidlo povolení k jízdě, příp. posunu. Pomocí družicové navigace je kontrolována práce strojvedoucího. Pokud se vozidlo blíží k místu konce povolení, je strojvedoucí varován. V případě, že vozidlo přejede místo povolení a určitou chvíli pojedou v nedovolené oblasti, Radioblok RB1 si začne být jistý, že je vozidlo tam, kde být nemá a vyvolá tak nouzovou brzdu. Ve verzi RB0+ podává GNSS pouze doplňkovou informaci, avšak součástí verze RB1 již bude na systému GPS založena bezpečnost, i když ne zcela. Pro přenos informací se využívá mobilní síť veřejného operátora s využitím přenosu dat GPRS. Činnost tohoto Radiobloku byla právě předmětem reálného testování, jež je popsáno v následujícím textu.

### **Faktory ovlivňující přesnost určení odhadu polohy**

Při implementaci družicové navigace v informačních a zejména bezpečnostních systémech v železniční dopravě je v první řadě nutné vypořádat se s několika faktory, které mají vliv na přesnost určení polohy. Přesnost bývá obecně degradována atmosférickými podmínkami, tedy ionosférou a troposférou. Rovněž záleží na geometrickém uspořádání družic (GDOP), což je uvedeno ve druhé kapitole této práce. Na železniční trati bývá příjem signálu ovlivněn bezprostředním okolím. Častým problémem je ztráta signálu způsobená hustým lesním porostem, budovami, zářezy či tunely. Kvůli okolním objektům může také docházet k odrazu signálu a následného příjmu odraženého signálu (tzv. multipath). Na tratích s elektrickým provozem může docházet rovněž k elektromagnetické interferenci. Ta bývá také důsledkem vysílání z externích zdrojů (vysílání v oblasti VHF), ale i z interních zdrojů vozidla (budicí obvody trakčních motorů, komunikačních zařízení apod.).

Nemělo by se zapomenout na vnější hrozby, jejichž cílem je zmatení uživatele a mezi které patří záměrné rušení signálů (angl. jamming), vysílání falešných navigačních signálů (angl. spoofing) a vysílání již přijatých signálů směrem k uživateli (angl. meaconing).

Není pochyb o tom, že velmi důležité je mít k dispozici kvalitní mapové podklady, tzv. digitální železniční mapy. Pokud nebude mapový podklad dostatečně přesný, může být chyba v určení polohy minimální, ale přesto nebude možné využít GNSS pro aplikace spojené s bezpečností železničního provozu.

Vliv ionosférické refrakce lze velmi dobře redukovat příjmem signálu na více frekvencích, nebo aplikováním relativně přesného modelu ionosféry. Z modelu pro troposféru lze rovněž stanovit lokální vliv troposféry opět s cílem redukce jejího vlivu. Účinky mnohacestného šíření signálu lze do jisté míry redukovat zejména použitím kvalitních antén přijímače uzpůsobených k potlačení tohoto jevu. Nežádoucí interference může být potlačena zesílením a filtrací signálů ještě před vstupem do navigačního přijímače.

## 4.2 Návrh postupu reálného měření

Pro ověření metody určení odhadu polohy přímo v terénu je nezbytné před samotným experimentem vytvořit jakousi metodiku, tedy návrh postupu s jednotlivými kroky seřazenými chronologicky za sebou. Obecný postup aplikovaný na konkrétní případ, jenž je uveden níže, byl sestaven autorem této práce na základě intuitivního přístupu formou řečnických otázek.

### Proč by měl být proveden reálný experiment?

Toto je základní otázka u každého reálného testování, přičemž je již zodpovězena v úvodu této kapitoly. Ačkoli simulace umožňují nesčetné možnosti nastavení, jde vždy pouze o napodobení skutečnosti. Simulace se chovají dle předem vytvořeného modelu, který byl naprogramován na základě zkušeností z praxe a pokud nejsou vhodně nastaveny parametry tohoto modelu nebo se uvažují jistá zjednodušení, nemusí výsledky odpovídat skutečnosti. Z těchto důvodů je provedení reálného experimentu nutné nejen pro ověření samotného předmětu testování, ale i modelu simulace.

### Co bude předmětem testování?

Jak již z názvu této kapitoly vyplývá, v rámci reálného experimentu budou ověřovány metody pro výpočet odhadu polohy. Prakticky půjde o testování matematických algoritmů, které jsou vytvořeny ve výpočetním programu MATLAB a které budou implementovány do GPS přijímače. Pro tento případ půjde o Bancroftovu algebraickou metodu, Newtonovu iterační metodu a rozšířený Kalmanův filtr.

### Co se od testování očekává?

Hypotézy a cíle reálného experimentu jsou totožné jako v případě testování na základě simulací. Smyslem je porovnat tyto tři metody z hlediska přesnosti poskytnutého odhadu a rovněž posoudit reakci na poruchy v přijatých datech. Cílem je tedy zjistit, zdali přesnosti daných metod jsou v reálu podobné přesnostem získaným na základě simulací a také ověřit, že lze tyto metody aplikovat i na reálná data.

### **Jaké bude místo a jaké budou podmínky pro testování?**

Velmi důležité je stanovení oblasti, kde bude testování probíhat. Vždy by se mělo vycházet ze znalosti účelu daného GPS přijímače. Vezme-li se fakt, že přijímač bude sloužit k určení polohy drážního vozidla na železniční trati, měl by být na železnici také otestován.

Pokud není předem známa konkrétní lokalita, měla by být pro účely testování vybrána trať, jejíž okolí je rozmanité (hustý lesní porost, tunely, mosty, městská zástavba). Mělo by být rovněž zohledněno roční období, a to zejména pokud se podél trati vyskytují opadavé stromy, které by v letních měsících bránily průchodu signálu na rozdíl od zimního období.

Co se týče podmínek samotného experimentu, bývají simulace vhodnější kvůli mnoha možnostem nastavení podmínek při testování. V případě měření v terénu je nezbytné počkat, až požadované podmínky okolního prostředí nastanou, případně provést testování v jiné lokalitě, kde ale nemusí být splněny ostatní podmínky týkající se požadovaného okolí podél trati apod.

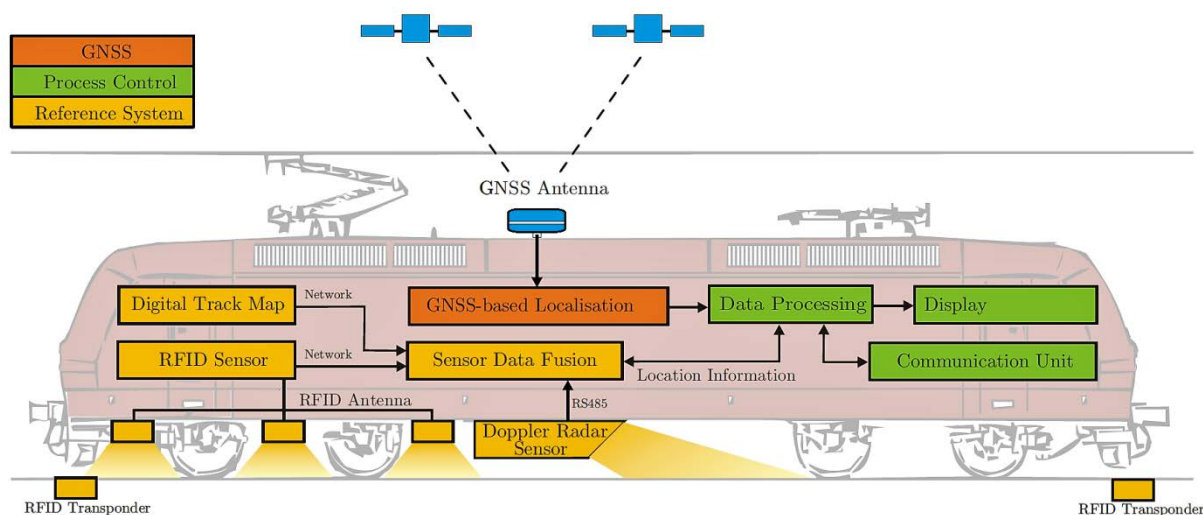
### **Co bude zapotřebí k provedení testu?**

Pro ověření metod určení odhadu polohy je pochopitelně nutný GPS přijímač s interní, nebo externí anténou, který poskytne RAW data (pseudovzdálenosti, efemeridy apod.). Dané algoritmy pro výpočet odhadu polohy budou poté na tato data aplikovány. V případě přijímače určeného pro lokalizaci na železniční síti je nezbytné zajistit drážní vozidlo a příslušnou trať včetně veškerého potřebného personálu (strojvedoucí, technik, IT expert) a vybavení. Pro případ následného vyhodnocení metod bude nutné obstarat záznamové médium, na které budou data ukládána. Pro zajištění správné činnosti během experimentu je vhodné propojit GPS přijímač s grafickým uživatelským rozhraním, díky kterému bude možné v reálném čase sledovat, zdali přijímač vypočítává odhad polohy, případně zjistit výpadek signálu apod.

Při testování by se rovněž mělo zohlednit, je-li cílem provádět testování pouze s družicemi GPS, případně s družicemi jiných GNSS (GLONASS, Galileo, COMPASS). Dle toho bude nutné obstarat GPS přijímač, nebo multikonstelační GNSS přijímač.

Jestliže bude cílem reálného měření zhodnotit přesnost metod (tedy velikosti chyb), pomocí nichž je určen odhad polohy, je nezbytné znát skutečnou polohu objektu. Stanovení skutečné polohy závisí na tom, zdali jde o statický, nebo dynamický objekt. V případě stojícího drážního vozidla na trati se nabízí geodetické měření, při kterém jsou využity finančně poměrně nákladné geodetické GNSS přijímače, které dosahují centimetrové až milimetrové přesnosti. Na území ČR je možné také využít sít permanentních stanic CZEPOS, které poskytují korekční data pro přesné určení polohy.

V případě stanovení polohy vozidla při jízdě je již nutné vybavit samotné vozidlo, případně trať senzory a detektory. Pro detekci vozidla by mohly být využity kolejové obvody, případně počítače náprav, avšak tyto systémy podají informaci pouze o oblasti, kde se vozidlo nachází a jejich použití tedy není vhodné. Při testování na železniční síti bývají obvykle aplikovány na trať a vozidlo RFID zařízení. Na tratích vybavených systémem ETCS lze pro stanovení polohy využít balízy umístěné v kolejišti. Vozidla jsou obvykle vybavena odometrem, avšak u nich je zapotřebí počítat s jistými chybami kvůli možnému prokluzování dvojkolí, případně že senzor nezachytí otáčku dvojkolí (chyby měření). Pro stanovení okamžité rychlosti může být na spodní části vozidla umístěn tzv. Dopplerův radar. Obrázek č. 42 zachycuje vybavení drážního vozidla určeného pro testování v terénu.



Obr. 42: Vybavení drážního vozidla pro testování v terénu [63, upraveno]

### Jakým způsobem a jak dlouho bude test prováděn?

Také samotné testování bude probíhat odlišně dle toho, zdali půjde o statický nebo dynamický objekt. Pokud bude testování probíhat pro jízdu drážního vozidla, bude nutné projet trať vícekrát, a to nejlépe oběma směry. V případě testování metod pro výpočet polohy nepohybujícího se objektu bude experiment probíhat po určitou dobu na předem definovaném místě trati. V obou případech ale platí, že pro relevantní výsledky bude nutné provést měření v terénu více než jednou (dynamický objekt), případně po dostatečně dlouhou dobu (statický objekt).

Neméně důležité je také stanovení intervalu měření. Pro statický objekt může být tento interval poměrně delší (řádově v minutách či desítkách minut), ovšem pro odhad polohy drážního vozidla za jízdy by interval měření měl být v řádu jednotek sekund. Stejně jako při testování na základě simulací i zde by bylo vhodné provést měření několik hodin denně (nejlépe 24

hodin kvůli různé konstelaci družic během dne) po dobu několika desítek dní. Ne vždy je ale možné zajistit stejný úsek trati pro tentýž čas a pro každý den v roce.

### Jaké budou vstupní podmínky?

Jak Newtonova iterační metoda, tak rozšířený Kalmanův filtr vyžaduje pro svou činnost počáteční podmínku pro odhad polohy přijímače. Pro výpočet dle Kalmanova filtru je navíc nutné předem stanovit počáteční kovarianci. V předešlém textu je již uvedeno, jakým způsobem probíhá stanovení počáteční podmínky pro polohu přímo pro testování v železniční dopravě. Počáteční podmínka by mohla být rovněž určena pomocí lokalizace dle základnových stanic BTS, díky čemuž lze získat představu o tom, ve které oblasti se objekt nachází.

Jestliže je požadováno provést klasifikaci výsledných odhadů dle hodnoty DOP, bude nutné tyto hodnoty nejdříve vypočítat. Nabízí se dva způsoby řešení. První z nich je totožný s tím, který byl proveden v simulacích, tedy že bude známa skutečná poloha objektu (GPS přijímače) a budou již známé pozice družic pro daný časový okamžik. Ovšem v případě neznalosti skutečné polohy testovaného objektu nebo pozic družic bude nutné hodnotu DOP stanovit dle jedné z metod popsaných v kapitole 3.1.1.

### Jaká data budou výstupem a v jakém budou formátu?

Aby výstupní data mohla být vyhodnocena, je pochopitelně nutná znalost struktury těchto dat včetně toho, v jakém formátu budou poskytnuta. V oblasti družicové navigace obecně platí, že data mohou být při měření předávána koncovému zařízení v reálném čase pomocí komunikačního protokolu (standardu) NMEA 0183, nebo mohou být ukládána na záznamové médium připojené k GPS přijímači ve formátu RINEX pro následné vyhodnocení. Obrázek níže zachycuje strukturu několika vět standardu NMEA 0183 (vlevo) a část dat ve formátu RINEX (vpravo). Obrázek je uveden pro představu o struktuře, a proto není detailněji popsán. [64]

```

$GPGGA,184050.84,3907.3839,N,12102.4772,W,1,05,1.8,00543,M,,,,*33
$GPRMC,184050.84,A,3907.3839,N,12102.4772,W,0.0,000.0,080301,15,E*54
$GPGSA,A,3,24,07,09,26,05,,,,,,,,,03.6,01.8,03.1*05
$PMGNST,02.12,3,T,534,05.0,+03327,00*40
$GPGLL,3907.3839,N,12102.4771,W,184051.812,A*2D
$GPGGA,184051.81,3907.3839,N,12102.4771,W,1,05,1.8,00543,M,,,,*34
$GPRMC,184051.81,A,3907.3839,N,12102.4771,W,0.0,000.0,080301,15,E*53
$GPGSA,A,3,24,07,09,26,05,,,,,,,,,03.6,01.8,03.1*05
$GPGSV,3,1,08,07,57,045,43,09,48,303,48,04,44,144,,02,39,092,*7F
$GPGSV,3,2,08,24,18,178,44,26,17,230,41,05,13,292,43,08,01,147,*75
$GPGSV,3,3,08,,,,,,,,,,,,,*71
$GPGLL,3907.3840,N,12102.4770,W,184052.812,A*21

```

```

2 OBSERVATION DATA M (MIXED) RINEX VERSION / TYPE
BLANK OR G = GPS, R = GLONASS, T = TRANSIT, H = MIXED COMMENT
XXRINEX0 V9.9 AUB 22-APR-93 12:43 PGM / RUN BY / DATE
EXAMPLE OF A MIXED RINEX FILE COMMENT
A 9080 MARKER NAME
9080.1.34 MARKER NUMBER
BILL SMITH ABC INSTITUTE OBSERVER / AGENCY
X1234A123 XX ZZZ REC # / TYPE / VERS
234 YY ANT # / TYPE
4375274. 587466. 4589095. APPROX POSITION XYZ
.9030 .0000 .0000 ANTENNA: DELTA H/E/N
1 1 WAVELENGTH FACT L1/2
1 2 6 G14 G15 G16 G17 G18 G19 WAVELENGTH FACT L1/2
4 P1 L1 L2 P2 # / TYPES OF OBSERV
18 INTERVAL
1990 3 24 13 10 36.0000000 TIME OF FIRST OBS
END OF HEADER
90 3 24 13 10 36.0000000 0 36126 9G 6 --.123456789
23629347.915 .300 8 23629364.158
20891534.648 -.120 9 -.358 20891541.292
20607600.189 -.430 9 .394 20607605.848
90 3 24 13 10 50.0000000 4 3
1 2 2 G 9 G12 WAVELENGTH FACT L1/2
*** WAVELENGTH FACTOR CHANGED FOR 2 SATELLITES *** COMMENT
90 3 24 13 10 54.0000000 0 56126 9G 6R21R22 --.123456789
23619095.458 -53875.632 8 -41981.375 23619112.088
20886075.667 -28688.027 9 -22354.535 20886082.101
20611072.689 10247.789 9 14219.770 20611078.410
21345678.576 12345.567 5
22123456.789 23456.789 5
90 3 24 13 11 0.0000000 2

```

Obr. 43: Standard NMEA 0183 a formát RINEX [65, 66]

Protokol NMEA 0183 umožňuje vysílání až 25 typů textových vět, jež se týkají geografické polohy přijímače, počtu dostupných družic, hodnot DOP a mnoho dalšího. Formát RINEX obsahuje veškerá data, která GNSS přístroj během observace získá. V souvislosti s výstupem dat je nutné dát si pozor na referenční soustavu a souřadnicové systémy, tedy zdali jsou data poskytnuta v kartézských či geodetických souřadnicích.

### **Jakým způsobem bude provedeno vyhodnocení?**

Vyhodnocení dat již probíhá obdobným způsobem jako při testování na základě simulací. Pro vyhodnocení je také velmi důležité znát parametry šumu. Výstupem měření budou odhady polohy objektu a díky známé skutečné poloze bude možné posoudit velikosti chyb daných metod a porovnat jejich přesnost. V případě, kdy se zjistí cokoli neočekávaného, bude nutné provést analýzu reálných dat a zjistit příčinu a poté opět provést simulaci, ve které již bude tato příčina zohledněna.

## **4.3 Provedení reálného měření**

Dne 26. října 2016 proběhlo reálné měření v rámci projektu RegioSAT společností AŽD Praha s.r.o. na úseku trati Číčenice – Volary (trať č. 197), a to za účelem demonstrace činnosti Radiobloku s označením RB1, jehož princip je popsán v předešlém textu. Na níže uvedeném obrázku je zachycen měřicí vůz společnosti AŽD Praha s.r.o. ve stanici Prachatice.



Obr. 44: Měřicí vůz [foto: Martin Pichl]

Cílem projektu RegioSAT je zvýšení bezpečnosti železničního provozu na regionálních tratích s využitím družicové navigace. Zvýšení bezpečnosti lze docílit přidáním k předchozí verzi Radiobloku RB0+ funkci bezpečného odhadu polohy a rychlosti. Rovněž jde o přidání bezpečné bodové kontroly rychlosti dle statického jízdního profilu (např. přes přejezdy), zautomatizování některých úkonů strojvedoucího a zavedení kontroly celistvosti vlaku. Obrázek č. 45 zachycuje stanoviště strojvedoucího spolu s obslužným rozhraním DMI a přenosným počítačem, jenž sloužil pouze k demonstrativnímu účelu této jízdy. Fotografie je pořízena při jízdě na trati ze stanice Čičenice směrem na Volary.



Obr. 45: Stanoviště strojvedoucího [foto: Lukáš Diblík]

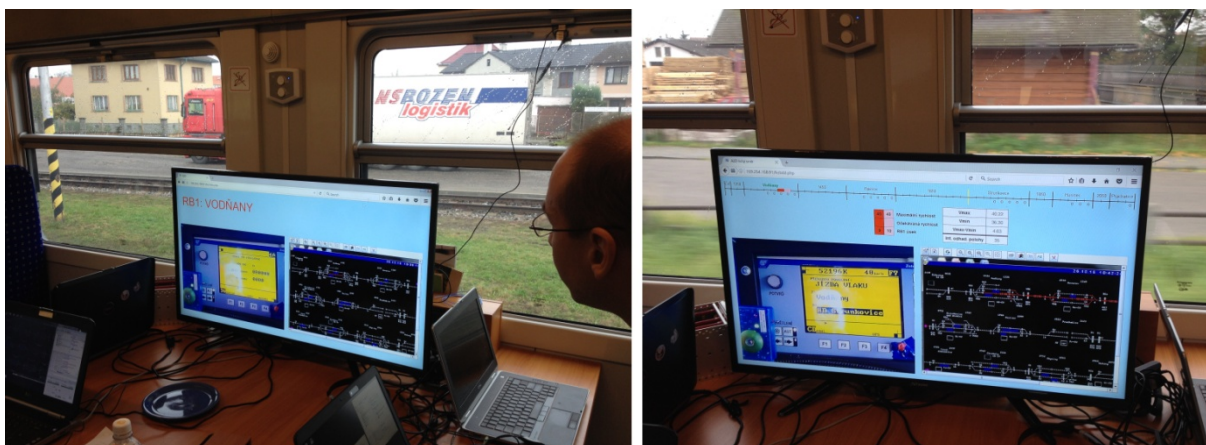
Níže uvedený obrázek zachycuje mobilní část Radiobloku, přičemž na pravé fotografii je zachycen GPS přijímač.



Obr. 46: Mobilní část Radiobloku (vlevo) a GPS přijímač (vpravo) [foto: Lukáš Diblík]



V případě dostupnosti bezpečných odhadů bude možné využívat nové funkcionality Radiobloku RB1. Avšak nebude-li bezpečný odhad poskytován, bude RB1 fungovat jako RB0+, přičemž o této situaci bude strojvedoucí neprodleně informován. Stacionární část Radiobloku při této jízdě byla zastoupena osobou s přenosným počítačem přímo v měřicím voze. Tato osoba udělovala povolení k jízdě do následujícího traťového oddílu. Obrázek č. 47 zachycuje kabinu pro cestující přizpůsobenou pro dané měření v terénu. Na televizní obrazovce byl promítán odhad aktuální polohy měřicího vozu spolu s odhadovanou okamžitou rychlostí. Rovněž byl k dispozici aktuální záběr na DMI radioblokového terminálu umístěného v kabině strojvedoucího a zobrazení schémat železničních stanic, které má ve skutečnosti k dispozici dispečer příslušné dopravní.



Obr. 47: Zobrazení informace o poloze měřicího vozu v reálném čase [foto: Lukáš Diblík]

Testování proběhlo na dané trati oběma směry a výstupy z měření byly použity pro zhodnocení metody pro výpočet odhadu polohy, která je v současné době vyvíjena společností AŽD Praha s.r.o.

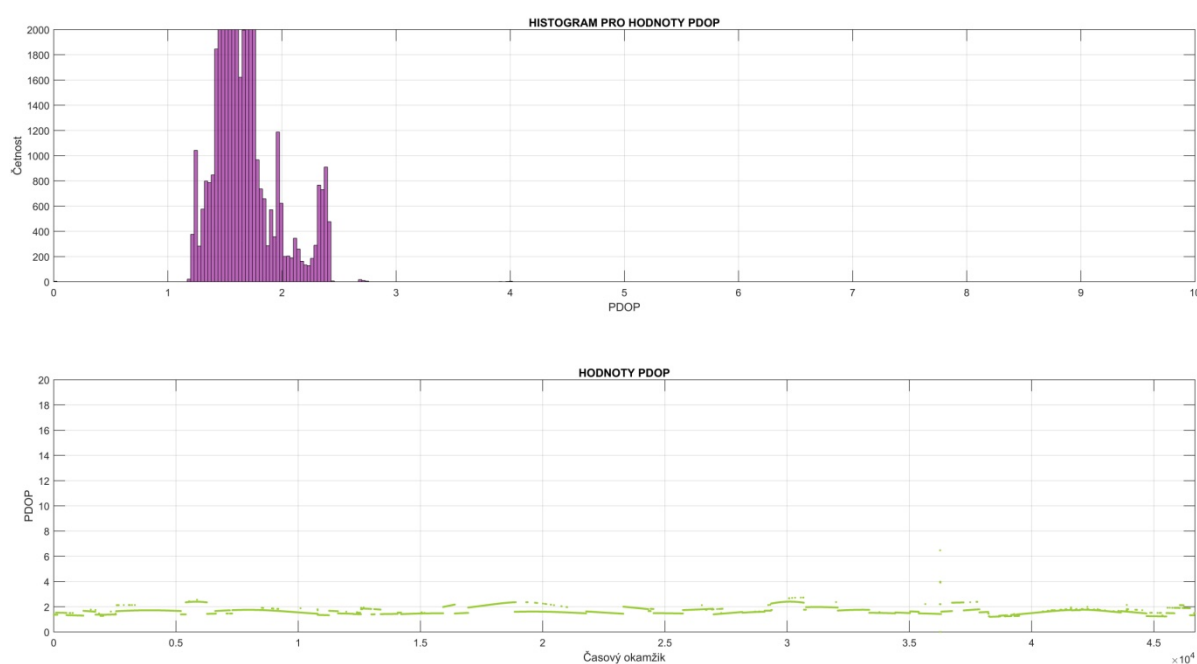
#### 4.4 Vyhodnocení výstupů reálného měření

Pro ověření, zdali příslušné matematické algoritmy pro výpočet odhadu polohy jsou schopny pracovat také se skutečně naměřenými daty, byla provedena ještě jedna simulace, která je níže popsána. Pro tato data byl pouze poupraven soubor Skript\_Algoritmus.m, nicméně nebylo nutné provádět složité úpravy v samotných funkcích díky tomu, že z naměřených dat byl odstraněn vliv atmosféry, troposféry, vliv transformace zohledňující rotaci Země apod. (simulace9). V těchto datech se ovšem vyskytují i časové okamžiky, kdy jsou k dispozici méně než čtyři družice, proto byla nastavena podmínka výpočtu pouze pro okamžiky, ve kterých jsou více než tři družice.

Opět je zde referenční polohou střecha budovy sídla společnosti AŽD Praha, s.r.o, přičemž data byla naměřena dne 5. října 2010. Tato data zahrnují měření po dobu 13 hodin s periodou jedné sekundy.

Testování metod probíhá na především základě simulačních dat. To je stěžejní, proto testy s naměřenými daty jsou spíše pro ujištění, zdali metody pracují stejným způsobem a poskytují obdobné výstupy jako při testování s daty vygenerovanými.

Na následujícím obrázku je zachycen histogram pro hodnoty PDOP a pod ním je umístěn graf závislosti PDOP na časových okamžicích. Z těchto grafů je patrné, že se zde nevyskytují hodnoty PDOP vyšší než 3.



Obr. 48: Histogram hodnot PDOP a hodnoty PDOP dle časových okamžiků (simulace9)

Obrázek č. 49 zachycuje čtyři grafy, kde je zobrazen časový průběh chyby odhadu polohy pro všechny metody, přičemž lze stanovit, že průběh všech analyzovaných metod je značně podobný a opět lze říci (na základě grafu na obrázku č. 48), že je ovlivněn hodnotou PDOP.

Graf na obrázku č. 50 reprezentuje maximální, střední a minimální hodnoty chyb odhadu stanovených dle všech příslušných metod. V případě, kdy není vidět bod, který reprezentuje danou metodu, znamená to, že je překryt bodem jiné metody.

Lze konstatovat, že pro relativně malé hodnoty PDOP (1 až 3) je přesnost všech těchto metod velmi podobná. V případě vyšších hodnot PDOP se již přesnost rozchází, i když pouze v rámci desetin metru, jak je vidět na grafu týkajícího se střední hodnoty chyb.

## Závěr

Na základě provedených simulací bylo možné zhodnotit analyzované matematické algoritmy pro výpočet odhadu polohy.

Bancroftova algebraická metoda podává velmi podobné výsledky jako Newtonova iterační metoda. Přestože Bancroftova metoda skutečně poskytuje přesnější odhad polohy v případě zhoršeného geometrického rozmístění družic, je rozdíl chyb těchto odhadů v řádech desetiny jednoho metru, tudíž s ohledem na lokalizaci vozidel na železniční síti lze říci, že přesnost obou metod je ve své podstatě stejná.

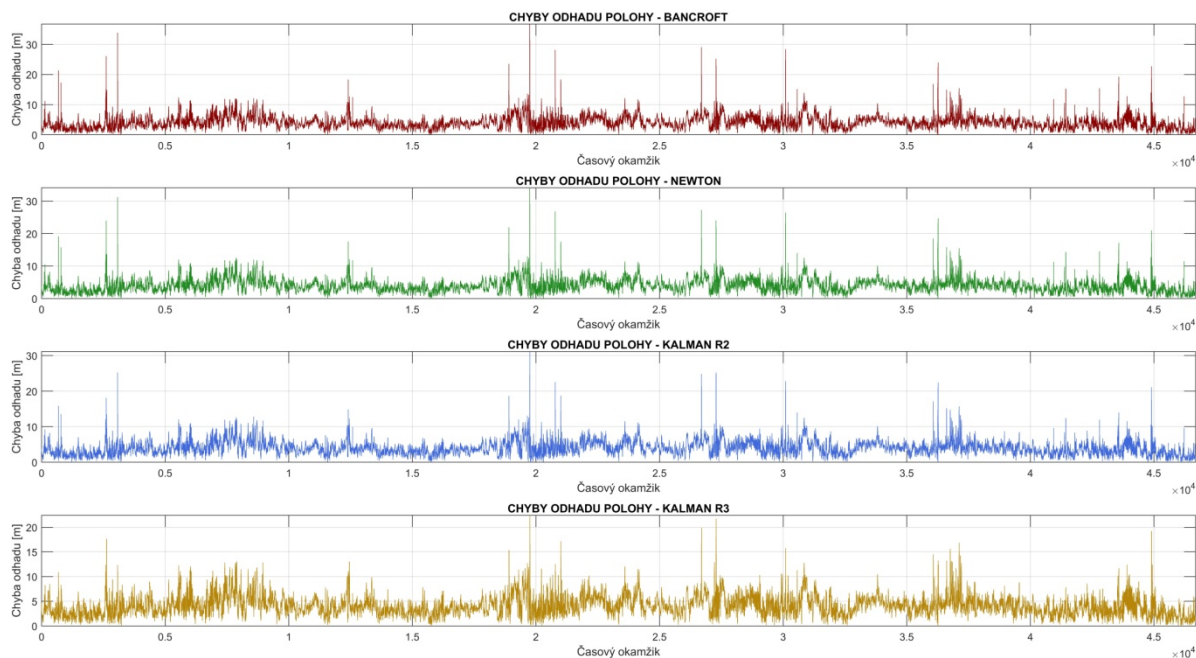
Ve všech provedených simulacích poskytoval Kalmanův filtr odhady polohy s nejmenšími chybami a rozptylem. Ideální je použití Kalmanova filtru pro třetí řád modelu. Zároveň byl prokázán nepříliš významný vliv geometrického uspořádání družic na přesnost Kalmanovy filtrace. Simulace také dokázaly opožděnou reakci Kalmanova filtru na přítomnost poruchy a rovněž jisté zpoždění této metody po odeznění poruchy.

Dále byly posouzeny vlivy různých kombinací STD šumu a injektované poruchy na přesnost odhadu polohy dle daných algoritmů. Bylo prokázáno, že hodnota STD šumu sice ovlivní přesnost odhadu, nicméně v případě, kdy nepřítomnosti poruchy není ovlivněna procentuální náležitost vůči kovariančnímu elipsoidu.

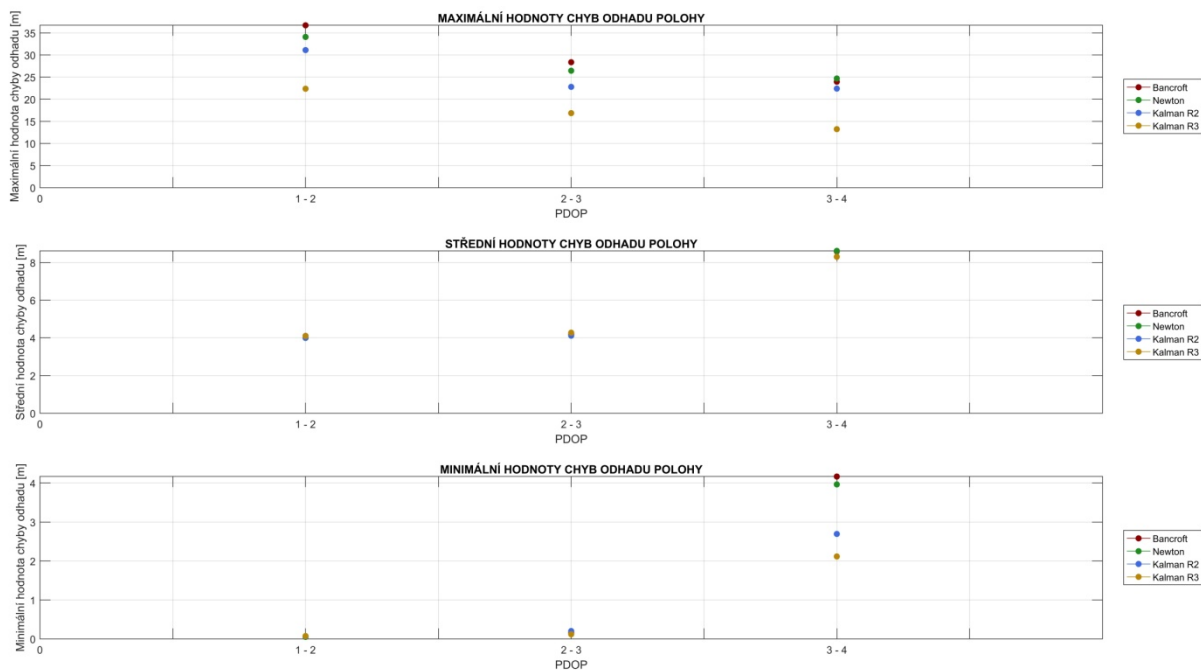
Nakonec bylo stanoveno, že odhad polohy vypočtený pomocí Bancroftovy a Newtonovy metody se pohybuje kolem referenční polohy přijímače nestranně, na rozdíl od Kalmanova filtru, u kterého je odhad polohy ve směru  $Y$  ovlivněn mnohem více než ve směrech  $X$  a  $Z$ , což může souviset s tím, že GPS je v odhadu nadmořské výšky méně přesná než u odhadu zeměpisné šířky a délky, přičemž v oblasti střední Evropy se tato výška projevuje spíše v  $X$  a  $Z$  než v  $Y$ .

Závěrem bych se osobně přiklonil k použití stanovení odhadu polohy na základě Kalmanova filtru pro model třetího řádu, jehož implementace je značně výhodná oproti zbylým analyzovaným metodám.

Pevně věřím, že tato práce bude přínosem nejen pro společnost AŽD Praha s.r.o., ale i pro odbornou veřejnost, jež se daným tématem zabývá.



Obr. 49: Chyby odhadu polohy zobrazené separátně (simulace9)



Obr. 50: Chyby odhadu polohy podle PDOP (simulace9)

Tato simulace s reálnými daty prokázala, že simulace s reálnými daty poskytuje shodné výstupy se simulacemi pracujícími s daty vygenerovanými.

# Seznam obrázků

Obr. 1: Fyzikální model povrchu Země - geoid [6, 7].....	13
Obr. 2: Kartézské a geodetické souřadnice [2, upraveno] .....	14
Obr. 3: Čas TAI, GPST a UTC [10, 11, upraveno] .....	15
Obr. 4: Princip určení polohy pomocí čtyř družic [13, 14].....	17
Obr. 5: Princip generování kódu C/A [24, upraveno].....	20
Obr. 6: Princip generování signálů GPS [25, upraveno] .....	22
Obr. 7: Princip modulace navigační zprávy a kódu [25, upraveno] .....	22
Obr. 8: Struktura navigační zprávy GPS [26, upraveno].....	23
Obr. 9: Modulace BPSK [25, upraveno] .....	24
Obr. 10: Synchronizace přijatého signálu s jeho kopií [8] .....	26
Obr. 11: Diskriminátor zpoždění [19, upraveno].....	27
Obr. 12: Autokorelační funkce [8] .....	28
Obr. 13: Správnost a přesnost [29, upraveno] .....	29
Obr. 14: Iterační proces odhadu polohy [8].....	35
Obr. 15: Kalmanův filtr obecné funkce [43, upraveno].....	38
Obr. 16: Různé elevační úhly družice [14, upraveno] .....	40
Obr. 17: Vliv rozmístění družic na přesnost odhadu polohy [2].....	42
Obr. 18: Kovarianční elipsoid a kovarianční elipsa [48, 49].....	45
Obr. 19: Normální rozdělení náhodné veličiny s vyznačenými hladinami významnosti [50] .....	45
Obr. 20: Zobrazení referenční polohy přijímače [51].....	46
Obr. 21: Vývojový diagram průběhu simulace [vlastní tvorba] .....	47
Obr. 22: Histogram hodnot PDOP a hodnoty PDOP dle časových okamžiků.....	54
Obr. 23: WGS 84 a dráhy družic .....	55
Obr. 24: Chyby odhadu polohy (simulace1).....	56
Obr. 25: Chyby odhadu polohy zobrazené separátně (simulace1) .....	56
Obr. 26: Chyby odhadu polohy podle PDOP (simulace1).....	57
Obr. 27: Chyby odhadu polohy podle PDOP – detail (simulace1).....	58
Obr. 28: Chyby odhadu polohy (simulace2).....	58

---

Obr. 29: Procentuální náležitost vůči kovariančnímu elipsoidu (vlevo sim.1, vpravo sim.2) .....	59
Obr. 30: Chyby odhadu polohy (nahore sim.3, dole sim.4).....	60
Obr. 31: Chyby odhadu polohy (nahore sim.5, dole sim.6).....	60
Obr. 32: Střední hodnoty chyb odhadu polohy (shora dolů sim.3 až sim.6) .....	61
Obr. 33: Procentuální náležitost vůči kovariančnímu elipsoidu (vlevo sim.4, vpravo sim.6) .....	63
Obr. 34: Odmocnina stopy KM (nahore sim.4, dole sim.5) .....	63
Obr. 35: Chyby odhadu polohy (nahore sim.7, dole sim.8).....	64
Obr. 36: Chyby odhadu polohy – detail (simulace8).....	65
Obr. 37: Chyby odhadu polohy – detail (simulace8).....	65
Obr. 38: Kategorie drah na území ČR [53, upraveno].....	68
Obr. 39: Kolejové obvody a počítače náprav [54, upraveno] .....	68
Obr. 40: Infrastrukturní a mobilní část systému ETCS [59, 60].....	69
Obr. 41: Úrovně systému ETCS [61] .....	70
Obr. 42: Vybavení drážního vozidla pro testování v terénu [63, upraveno].....	75
Obr. 43: Standard NMEA 0183 a formát RINEX [65, 66].....	76
Obr. 44: Měřicí vůz [foto: Martin Pichl] .....	77
Obr. 45: Stanoviště strojvedoucího [foto: Lukáš Diblík] .....	78
Obr. 46: Mobilní část Radiobloku (vlevo) a GPS přijímač (vpravo) [foto: Lukáš Diblík].....	78
Obr. 47: Zobrazení informace o poloze měřicího vozu v reálném čase [foto: Lukáš Diblík] .....	79
Obr. 48: Histogram hodnot PDOP a hodnoty PDOP dle časových okamžiků (simulace9).....	80
Obr. 49: Chyby odhadu polohy zobrazené separátně (simulace9) .....	81
Obr. 50: Chyby odhadu polohy podle PDOP (simulace9).....	81

# Seznam tabulek

Tab. 1: Tabulka parametrů simulace .....	54
Tab. 2: Hodnoty chyb odhadu polohy (simulace4).....	62

# Seznam použité literatury

- [1] RAPANT, Petr. *Družicové polohové systémy* [online]. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2002 [cit. 2016-04-04]. ISBN 80-248-0124-8. Dostupné z: [http://gis.vsb.cz/vojtek/content/gnps/files/\\_source/RAP02.pdf](http://gis.vsb.cz/vojtek/content/gnps/files/_source/RAP02.pdf)
- [2] VOJTEK, David. *Globální navigační a polohové systémy* [online]. Ostrava: VŠB-TU Ostrava, 2014 [cit. 2016-04-04]. Dostupné z: [http://gis.vsb.cz/vojtek/content/gnps/files/\\_source/Ucebni-texty-GNPS-distancni.pdf](http://gis.vsb.cz/vojtek/content/gnps/files/_source/Ucebni-texty-GNPS-distancni.pdf)
- [3] KAPLAN, Elliott D a C HEGARTY. *Understanding GPS: principles and applications*. 2nd ed. Boston: Artech House, c2006. ISBN 15-805-3894-0.
- [4] MISRA, Pratap. *Global positioning system: signals, measurements, and performance*. 2nd ed. Lincoln, Mass.: Ganga-Jamuna, c2006. ISBN 0970954417.
- [5] EL-RABBANY, Ahmed. *Introduction to GPS: the Global Positioning System*. Boston, MA: Artech House, c2002. ISBN 15-805-3183-0.
- [6] *Phys.org* [online]. [cit. 2016-04-04]. Dostupné z: [http://cdn.phys.org/newman/gfx/news/hires/2012/c71\\_geoid\\_smooth4.jpg](http://cdn.phys.org/newman/gfx/news/hires/2012/c71_geoid_smooth4.jpg)
- [7] *Mathworks* [online]. [cit. 2016-04-04]. Dostupné z: <http://www.mathworks.com/help/map/understanding-spherical-coordinates.html?requestedDomain=www.mathworks.com>
- [8] ŠEBESTA, Jiří. *Globální navigační systémy* [online]. Vyd. 1. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2012 [cit. 2016-04-04]. ISBN 978-80-214-4500-0. Dostupné z: [http://www.urel.feec.vutbr.cz/~sebestaj/RAR/literatura/Globalni\\_navigacni\\_systemy.pdf](http://www.urel.feec.vutbr.cz/~sebestaj/RAR/literatura/Globalni_navigacni_systemy.pdf)
- [9] MACHOTKA, Radovan a Jan FIXEL. *Geodetická astronomie a kosmická geodézie II: Kosmická geodézie* [online]. Brno: VUT v Brně, 2007 [cit. 2016-04-04]. Dostupné z: [http://fast.darmy.net/opory%20-%20IV%20Mgr/HE01\\_M02-Geodetická%20astronomie.pdf](http://fast.darmy.net/opory%20-%20IV%20Mgr/HE01_M02-Geodetická%20astronomie.pdf)
- [10] *JJY* [online]. [cit. 2016-04-04]. Dostupné z: <http://jjy.nict.go.jp/mission/leapsecond-e8.jpg>
- [11] *Timeanddate.com* [online]. [cit. 2016-04-04]. Dostupné z: <http://www.timeanddate.com/time/international-atomic-time.html>
- [12] TSUI, James Bao-yen. *Fundamentals of global positioning system receivers: a software approach*. 2nd ed. New Jersey: John Wiley, c2005. ISBN 04-717-0647-7.
- [13] *Openclipart* [online]. [cit. 2016-04-04]. Dostupné z: <https://openclipart.org/download/191659/GPS-3D-trilateration.svg>
- [14] *Openclipart* [online]. [cit. 2016-04-04]. Dostupné z: <https://openclipart.org/download/191680/GPS-GDOP-4-good.svg>



- [15] *GPS.gov* [online]. [cit. 2016-04-04]. Dostupné z: <http://www.gps.gov/systems/gps/space/>
- [16] ŠVÁBENSKÝ, Otakar, Josef WEIGEL a Jan FIXEL. *Základy GPS a jeho praktické aplikace*. Brno: CERM, 1995. ISBN 80-214-0620-8.
- [17] HOFMANN-WELLENHOF, B, Herbert LICHTENEGGER a Elmar WASLE. *GNSS--global navigation satellite systems: GPS, GLONASS, Galileo, and more*. New York: Springer, c2008. ISBN 978-321-1730-126.;
- [18] LÁSKA, Zdeněk, Martin TEŠNAR, Jaroslav SLABÝ a Jan SUKUP. *Globální navigační satelitní systémy a jejich využití v praxi* [online]. Brno: VUT v Brně, 2010 [cit. 2016-04-04]. Dostupné z: [http://www.crr.vutbr.cz/system/files/brozura\\_08\\_1009.pdf](http://www.crr.vutbr.cz/system/files/brozura_08_1009.pdf)
- [19] HRDINA, Zdeněk, Petr PÁNEK a František VEJRAŽKA. *Rádiové určování polohy: Družicový systém GPS*. 1. vyd. Praha: ČVUT, 1995. ISBN 8001013863.
- [20] ČÁBELKA, Miroslav. *Úvod do GPS* [online]. [cit. 2016-04-04]. Dostupné z: [http://www.natur.cuni.cz/geografie/geoinformatika-kartografie/ke-stazeni/vyuka/gps/skriptum-uvod-do-gps?student\\_welcome=1](http://www.natur.cuni.cz/geografie/geoinformatika-kartografie/ke-stazeni/vyuka/gps/skriptum-uvod-do-gps?student_welcome=1)
- [21] ŠEBESTA, Jiří. *Radiolokace a radionavigace: přednášky*. Vyd. 1. Brno: Vysoké učení technické, 2004. ISBN 8021424826.
- [22] MENDIZÁBAL SAMPER, Jaizki, Roc BERENGUER a Juan MELÉNDEZ. *GPS & GALILEO: dual RF front-end receiver and design, fabrication, and test*. New York: McGraw-Hill, 2009. ISBN 00-715-9869-3.
- [23] HRDINA, Zdeněk a František VEJRAŽKA. *Digitální radiová komunikace*. Vyd. 2. Praha: České vysoké učení technické, 1998. ISBN 80-010-1820-2.
- [24] *Natronics* [online]. [cit. 2016-04-04]. Dostupné z: <https://natronics.github.io/blog/2014/gps-prn/>
- [25] *ILectureOnline* [online]. [cit. 2016-04-04]. Dostupné z: <http://www.ilectureonline.com/lectures/subject/SPECIAL%20TOPICS/31/235>
- [26] *GPS: Essentials of Satellite Navigation* [online]. u-blox, 2007 [cit. 2016-04-04]. Dostupné z: <http://www.u-blox.com/en/technology/GPS-X-02007.pdf>
- [27] GREWAL, Mohinder S, Lawrence R WEILL a Angus P ANDREWS. *Global positioning systems, inertial navigation, and integration*. 2nd ed. Hoboken: Wiley, 2006. ISBN 9780470099728.
- [28] VEJRAŽKA, František a Petr KAČMAŘÍK. *Galileo - systém družicové navigace* [online]. In: . [cit. 2016-04-04]. Dostupné z: <http://www.roznovskastredni.cz/dwnl/pel2011/02/vejrazka.pdf>
- [29] Smart Data Series: Accurate Data. *Exelate* [online]. New York: The Nielsen Company, 2014 [cit. 2016-04-17]. Dostupné z: <http://exelate.com/newsite/wp-content/uploads/2014/02/accuracy-precision.png>
- [30] Pokročilé metody zpracování GNSS měření. *Institut geoinformatiky VŠB* [online]. Ostrava, 2014 [cit. 2016-04-17]. Dostupné z: <http://gis.vsb.cz/kacmarik/pmzgnss/cv02/cv02.html>

- [31] ZWIRELLO, Lukasz. *Realization Limits of Impulse-Radio UWB Indoor Localization Systems*. Karlsruhe, Baden: KIT Scientific Publishing, 2013. ISBN 37-315-0114-7.
- [32] YANG, Ming a Kuo-Hwa CHEN. Performance Assessment of a Noniterative Algorithm for Global Positioning System (GPS) Absolute Positioning. *Proceedings of the National Science Council, Republic of China* [online]. 2001, **25**(2), 102-106 [cit. 2016-04-17]. ISSN 0255-6596. Dostupné z: <http://www1.geomatics.ncku.edu.tw/staff-1.php?download=58>
- [33] BANCROFT, Stephen. An Algebraic Solution of the GPS Equations. *IEEE Transactions on Aerospace and Electronic Systems* [online]. 1985, **21**(1), 56-59 [cit. 2016-04-17]. DOI: 10.1109/TAES.1985.310538. ISSN 0018-9251. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4104017>
- [34] PAJARES, M. Hernández, J. M. Juan ZORNOZA a J. Sanz SUBIRANA. *GPS data processing: code and phase: algorithms, techniques and recipes* [online]. 1st ed. (English). Barcelona: Centre de Publicacions del Campus Nord, UPC, 2005 [cit. 2016-04-17]. ISBN 84-932-2305-0. Dostupné z: [http://gage.upc.edu/sites/default/files/TEACHING\\_MATERIAL/GPS\\_BOOK/ENGLISH/PDGPS/BOOK\\_PDGPS\\_gAGE\\_NAV\\_08.pdf](http://gage.upc.edu/sites/default/files/TEACHING_MATERIAL/GPS_BOOK/ENGLISH/PDGPS/BOOK_PDGPS_gAGE_NAV_08.pdf)
- [35] SHUANGGEN, Jin. *Global navigation satellite systems: signal, theory and applications*. Rijeka, Croatia: Intech, 2012. ISBN 978-953-3078-434.
- [36] CHAFFEE, James a Jonathan ABEL. On the exact solutions of pseudorange equations. *IEEE Transactions on Aerospace and Electronic Systems* [online]. 1994, **30**(4), 1021-1030 [cit. 2016-04-17]. DOI: 10.1109/7.328767. ISSN 0018-9251. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=328767>
- [37] KUMAR, B. Hari. Estimation of GPS User Position Using Bancroft Algorithm. *SSRG International Journal of Electronics and Communication Engineering* [online]. 2015, **2**(4), 6-10 [cit. 2016-04-17]. ISSN 2348 – 8549. Dostupné z: <http://www.internationaljournalssrg.org/IJECE/2015/Volume2-Issue4/IJECE-V2I4P102.pdf>
- [38] RAHEMI, N., M. R. MOSAVI, A. A. ABEDI a S. MIRZAKUCHAKI. Accurate Solution of Navigation Equations in GPS Receivers for Very High Velocities Using Pseudorange Measurements. *Advances in Aerospace Engineering* [online]. 2014, **2014**(ID 435891), 1-8 [cit. 2016-04-17]. DOI: 10.1155/2014/435891. ISSN 2356-6531. Dostupné z: <http://www.hindawi.com/journals/aae/2014/435891/>
- [39] ZEKAVAT, Seyed A. a R. Michael BUEHRER. *Handbook of Position location: theory, practice and advances* [online]. Hoboken, N.J.: Wiley-IEEE Press, 2012 [cit. 2016-04-17]. ISBN 978-111-8104-774. Dostupné z: <http://onlinelibrary.wiley.com/book/10.1002/9781118104750>
- [40] HE, Y. a A. BILGIC. Iterative least squares method for global positioning system. *Advances in Radio Science* [online]. 2011, **9**, 203-208 [cit. 2016-04-17]. DOI: 10.5194/ars-9-203-2011. ISSN 1684-9973. Dostupné z: <http://www.adv-radio-sci.net/9/203/2011/>

- [41] DOBEŠ, Josef a Václav ŽALUD. *Moderní radiotechnika*. 1. vyd. Praha: BEN - technická literatura, 2006. ISBN 80-730-0132-2.
- [42] GIBBS, Bruce. *Advanced kalman filtering, least-squares and modeling: a practical handbook* [online]. Hoboken: John Wiley, c2011 [cit. 2016-04-24]. ISBN 978-0-470-52970-6. Dostupné z: <http://onlinelibrary.wiley.com/book/10.1002/9780470890042>
- [43] *The Extended Kalman Filter: An Interactive Tutorial for Non-Experts* [online]. [cit. 2016-04-24]. Dostupné z: [http://home.wlu.edu/~levys/kalman\\_tutorial/](http://home.wlu.edu/~levys/kalman_tutorial/)
- [44] ŠIMANDL, Miroslav. *Identifikace systémů a filtrace*. 1. vyd. - dotisk. Plzeň: Západočeská univerzita v Plzni, 2001. ISBN 80-708-2170-1.
- [45] GREWAL, Mohinder S. a Angus P. ANDREWS. *Kalman filtering: theory and practice using MATLAB* [online]. 3rd ed. Hoboken, N.J.: Wiley, c2008 [cit. 2016-04-24]. ISBN 04-701-7366-1. Dostupné z: <http://onlinelibrary.wiley.com/book/10.1002/9781118984987>
- [46] BROOKNER, Eli. *Tracking and Kalman filtering made easy* [online]. New York: Wiley, c1998 [cit. 2016-04-24]. ISBN 0-471-18407-1. Dostupné z: <http://onlinelibrary.wiley.com/book/10.1002/0471224197>
- [47] CHENG, Chao-heh. *Calculations for Positioning with the Global Navigation Satellite System*. Ohio, 1998. Thesis for Degree Master of Science. Ohio University.
- [48] *Mathground* [online]. [cit. 2016-10-30]. Dostupné z: <http://mathground.net/wp-content/uploads/2013/07/Ellipsoid3D.jpg>
- [49] *Computer Vision for Dummies* [online]. [cit. 2016-10-30]. Dostupné z: [http://www.visiondummy.com/wp-content/uploads/2014/04/error\\_ellipse\\_isocontours.png](http://www.visiondummy.com/wp-content/uploads/2014/04/error_ellipse_isocontours.png)
- [50] *Magoosh* [online]. [cit. 2016-10-30]. Dostupné z: <https://s3.amazonaws.com/magoosh-company-site/wp-content/uploads/mat/files/2016/06/07130639/Bell-Curve-1.png>
- [51] *2GIS - Mapa Prahy* [online]. [cit. 2016-10-30]. Dostupné z: <https://2gis.cz/praha>
- [52] *SŽDC* [online]. [cit. 2016-11-25]. Dostupné z: <http://www.szdc.cz/o-nas/zeleznice-cr/zeleznicni-sit-v-cr.html>
- [53] *SŽDC: Provozování dráhy* [online]. [cit. 2016-11-25]. Dostupné z: <http://provoz.szdc.cz/PORTAL/ViewArticle.aspx?oid=594598>
- [54] *Osobní stránky doc. Ing. Lukáše Týfky, Ph.D.* [online]. [cit. 2016-11-25]. Dostupné z: <https://www.fd.cvut.cz/personal/tyfal/>
- [55] TRAŤ Č. 197 VOLARY – ČÍČENICE – TRAŤ, KDE UMÍRALI CESTUJÍCÍ. *Drážní inspekce* [online]. [cit. 2016-11-25]. Dostupné z: <http://www.dicr.cz/trat-c-197-volary-cicenice-trat-kde-umirali-cestujici>

- [56] FILIP, Aleš a Jiří SUCHÁNEK. *Certifikace satelitního signálu GALILEO pro železniční aplikace* [online]. 22. Praha, 2006 [cit. 2016-11-25]. Dostupné z: <https://vts.cd.cz/documents/51448/51571/2603.pdf/dc48817b-f87b-4893-87e6-41c06bd9e158>
- [57] BRANDEJSKÝ, Tomáš, Martina VITTEKOVÁ, Jaroslav VAŠÁTKO, Jitka ŘEZNÍČKOVÁ a Martin KRÁL. *Satelitní navigace na železnici* [online]. Praha, 2015 [cit. 2016-11-25]. ISBN 978-80-214-5144-5. Dostupné z: <http://www.sizi.cz/file.php?nid=14068&oid=4665639>
- [58] FIKEJZ, Jan. Možnosti lokalizace kolejových vozidel v železniční síti. *Elektrorevue* [online]. 2012, 14(4), 8 [cit. 2016-11-25]. ISSN 1213-1539. Dostupné z: <http://elektrorevue.cz/file.php?id=200000856-ef015effb4>
- [59] Skúšky ETCS alebo s Taurusom na trati Rača - Veľké Kostoľany. *Vlaky.net* [online]. [cit. 2016-11-25]. Dostupné z: <http://www.vlaky.net/zeleznice/spravy/003024-Skusky-ETCS-alebo-s-Taurusom-na-trati-Raca-Velke-Kostolany/>
- [60] Systém ETCS zajistí vyšší bezpečnost železniční dopravy. *Dopravní noviny* [online]. 2013 [cit. 2016-11-25]. Dostupné z: <http://www.dnoviny.cz/assets/images/84b55879c3806455bc509781babc1e44/3013-640.jpg>
- [61] European Train Control System (ETCS). *THALES* [online]. [cit. 2016-11-25]. Dostupné z: <https://www.thalesgroup.com/fr/european-train-control-system-etcs>
- [62] FILIP, Aleš. *Využití satelitní navigace v železniční zabezpečovací technice* [online]. , 14 [cit. 2016-11-25]. Dostupné z: [http://www.railvolution.net/czechraildays/2005/seminare/i\\_8.pdf](http://www.railvolution.net/czechraildays/2005/seminare/i_8.pdf)
- [63] *ResearchGate* [online]. [cit. 2016-11-25]. Dostupné z: [https://www.researchgate.net/profile/Cyril\\_Legrand/publication/261693053/figure/fig1/AS:296906275213316@1447799380617/Figure-2-DemoOrt-Structure-2.png](https://www.researchgate.net/profile/Cyril_Legrand/publication/261693053/figure/fig1/AS:296906275213316@1447799380617/Figure-2-DemoOrt-Structure-2.png)
- [64] Komunikační protokoly a formáty. *Vysoká škola báňská - Technická univerzita Ostrava - VŠB-TUO* [online]. [cit. 2016-11-25]. Dostupné z: [http://gisak.vsb.cz/vojtek/content/gnps/files/\\_source/Ucebni-texty-GNPS-distančni.pdf](http://gisak.vsb.cz/vojtek/content/gnps/files/_source/Ucebni-texty-GNPS-distančni.pdf)
- [65] NMEA data. *GPSInformation.net* [online]. [cit. 2016-11-25]. Dostupné z: <http://www.gpsinformation.org/dale/nmea.htm#GSA>
- [66] RINEX: The Receiver Independent Exchange Format Version 2. *University of Colorado* [online]. [cit. 2016-11-25]. Dostupné z: <http://www.colorado.edu/engineering/ASEN/asen6090/rinex.format>

# Seznam příloh

## Příloha A: Skripty vytvořené v programu MATLAB

Skript_Algoritmus.m.....	I
Funkce_PDOP.m.....	XX
Funkce_Bancroft.m.....	XXI
Funkce_Newton.m.....	XXII
Funkce_KalmanR2.m.....	XXIII
Funkce_KalmanR3.m.....	XXIV
Funkce_Elipsoid.....	XXV

## Příloha B: Výstupy provedených simulací

Grafy společné pro všechny simulace.....	XXVI
Simulace 1.....	XXVII
Simulace 2.....	XXX
Simulace 3.....	XXXIV
Simulace 4.....	XXXVII
Simulace 5.....	XLI
Simulace 6.....	XLIV
Simulace 7.....	XLVIII
Simulace 8.....	LI
Simulace 9.....	LV

# Příloha A: Skripty vytvořené v programu MATLAB

```
% SKRIPT ALGORITMUS
% -----
% Hlavní skript, ve kterém jsou načtena data,
% definovány parametry, vloženy odkazy na funkce,
% vypočteny chyby odhadu, provedeno vyhodnocení
% a zobrazeny grafy
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016
```

## PŘÍPRAVA SIMULACE

```
clc % Vyčištění Command Window
clear all % Vymazání všech proměnných z Workspace
clear global % Vymazání globálních proměnných z Workspace
close all % Uzavření všech grafů
clock; % Funkce pro zobrazení dne a času
DateStart = fix(clock) % Zobrazení dne a času začátku simulace [RRRR MM DD hh mm ss]
tic % Počátek časovače (od této chvíle je počítána doba simulace)
load Data_CelyDen % Nahrání vstupních dat (poloha družic, poloha přijímače, pseudovzdálenosti...)
```

## PARAMETRY SIMULACE

```
PocetDnu = 365; % Počet simulovaných dnů
PocetOkamziku = 8000; % Počet simulovaných okamžiků
STDsumu = 2.5; % Směrodatná odchylka šumu
c = 299792458; % Rychlost šíření radiového signálu (čili rychlost světla ve vakuu [m/s])
S0 = [4061678.7767058, 1048478.06265762, 4789000.14428881, -5]; % Počáteční odhad souřadnic a diference GPST a hodin přijímače
T = 2; % Perioda vzorkování pro EKF
SVporucha = 29; % Označení družice, jež bude vysílat s poruchou
```

## POLE PRO VÝSLEDKY SIMULACE

```
CasOkamPDOP = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),1); % Pole NaN pro hodnoty PDOP
ChybyOdhaduBancroft = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu - Bancroft
ChybyOdhaduBancroft0saX = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu v ose X - Bancroft
ChybyOdhaduBancroft0saY = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu v ose Y - Bancroft
ChybyOdhaduBancroft0saZ = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu v ose Z - Bancroft
ChybyOdhaduNewton = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu - Newton
ChybyOdhaduNewton0saX = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu v ose X - Newton
ChybyOdhaduNewton0saY = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu v ose Y - Newton
ChybyOdhaduNewton0saZ = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu); % Pole NaN pro chyby odhadu v ose Z - Newton
```

```

ChybyOdhaduKalmanR2 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR20saX = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR20saY = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR20saZ = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR3 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR30saX = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR30saY = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
ChybyOdhaduKalmanR30saZ = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
HodnoceniKstopaNewton = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
HodnoceniKdeterminantNewton = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
HodnoceniKstopaKalmanR2 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
HodnoceniKdeterminantKalmanR2 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
HodnoceniKstopaKalmanR3 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
HodnoceniKdeterminantKalmanR3 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
Elipsoid68Newton = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
Elipsoid95Newton = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
Elipsoid68KalmanR2 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
Elipsoid95KalmanR2 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
Elipsoid68KalmanR3 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
Elipsoid95KalmanR3 = NaN*ones(length(KONSTELACE_Ivan.pocetdruzic),PocetDnu);
% Pole NaN pro chyby odhadu - KalmanR2
% Pole NaN pro chyby odhadu v ose X - KalmanR2
% Pole NaN pro chyby odhadu v ose Y - KalmanR2
% Pole NaN pro chyby odhadu v ose Z - KalmanR2
% Pole NaN pro chyby odhadu - KalmanR3
% Pole NaN pro chyby odhadu v ose X - KalmanR3
% Pole NaN pro chyby odhadu v ose Y - KalmanR3
% Pole NaN pro chyby odhadu v ose Z - KalmanR3
% Pole NaN pro stopu KM - Newton
% Pole NaN pro determinant KM - Newton
% Pole NaN pro stopu KM - KalmanR2
% Pole NaN pro determinant KM - KalmanR2
% Pole NaN pro stopu KM - KalmanR3
% Pole NaN pro determinant KM - KalmanR3
% Pole NaN pro kovarianční elipsoid 68 % - Newton
% Pole NaN pro kovarianční elipsoid 95 % - Newton
% Pole NaN pro kovarianční elipsoid 68 % - KalmanR2
% Pole NaN pro kovarianční elipsoid 95 % - KalmanR2
% Pole NaN pro kovarianční elipsoid 68 % - KalmanR3
% Pole NaN pro kovarianční elipsoid 95 % - KalmanR3

```

## VLASTNÍ ALGORITMUS SIMULACE

```

for k = 1:PocetDnu
    PolohaPrijimace = KONSTELACE_Ivan.PolohyRychlosti(k,2:4);

    X2 = NaN*ones(8,1);
    X2([1,3,5]) = [S0(1),S0(2),S0(3)];
    X2([2,4,6]) = [0,0,0];
    X2(7,1) = S0(4);
    X2(8,1) = 0;
    P2 = 10*eye(8);
    X3 = NaN*ones(12,1);
    X3([1,4,7]) = [S0(1),S0(2),S0(3)];
    X3([2,5,8]) = [0,0,0];
    X3([3,6,9]) = [0,0,0];
    X3(10,1) = S0(4);
    X3(11,1) = 0;
    X3(12,1) = 0;
    P3 = 10*eye(12);

    for t = 1:PocetOkamziku
        GenerSum = randn(KONSTELACE_Ivan.pocetdruzic(t),1)*STDsumu;
        PoziceDruzic = KONSTELACE_Ivan.druzice(KONSTELACE_Ivan.novycasokam(t):(KONSTELACE_Ivan.novycasokam(t+1)-1),3:5);
        % Cyklus FOR pro jednotlivé dny
        % Stanovení polohy X,Y,Z přijímače

        % Stavový vektor 2.řádu (State Vector) s NaN X = [x,vx,y,vy,z,vz,b,d]'
        % Počáteční odhad polohy přijímače
        % Počáteční odhad rychlosti přijímače
        % Počáteční odhad difference GPST a hodin přijímače
        % Počáteční odhad rychlosti difference hodin přijímače
        % Počáteční odhad kovarianční matice chyb stavu (State Error CovMat)
        % Stavový vektor 3. řádu (State Vector) s NaN X = [x,vx,ax,y,vy,ay,z,vz,az,b,d,m]'
        % Počáteční odhad polohy přijímače
        % Počáteční odhad rychlosti přijímače
        % Počáteční odhad zrychlení přijímače
        % Počáteční odhad difference GPST a hodin přijímače
        % Počáteční odhad rychlosti difference hodin přijímače
        % Počáteční odhad zrychlení difference hodin přijímače
        % Počáteční odhad kovarianční matice chyb stavu (State Error CovMat)

        % Vnořený FOR cyklus pro jednotlivé časové okamžiky
        % Generovaný šum s normálním rozdělením a směrodatnou odchylkou "STDsumu"
        % Přiřazení souřadnic X,Y,Z družic k jednotl. čas. okam.
    end
end

```

```

CasOkamPDOP(t) = Funkce_PDOP(PoziceDruzic,PolohaPrijimace(1),PolohaPrijimace(2),PolohaPrijimace(3));
PSR = KONSTELACE_Ivan.druzice(KONSTELACE_Ivan.novycasokam(t):(KONSTELACE_Ivan.novycasokam(t+1)-1),6)+...
    c*KONSTELACE_Ivan.druzice(KONSTELACE_Ivan.novycasokam(t):(KONSTELACE_Ivan.novycasokam(t+1)-1),7);
PSRsum = PSR+GenerSum;
SV = KONSTELACE_Ivan.druzice(KONSTELACE_Ivan.novycasokam(t):(KONSTELACE_Ivan.novycasokam(t+1)-1),2);
SVindex = find(SV == SVporucha);

%if t >= 4700 && t <= 4730
%PSRsum(SVindex) = PSRsum(SVindex)+50;
%end

[OdhadPolohyBancroft] = Funkce_Bancroft(PoziceDruzic,PSRsum);
ChybyOdhaduBancroft(t,k) = norm(OdhadPolohyBancroft(1:3)-PolohaPrijimace);
ChybyOdhaduBancroft0saX(t,k) = norm(OdhadPolohyBancroft(1)-PolohaPrijimace(1));
ChybyOdhaduBancroft0saY(t,k) = norm(OdhadPolohyBancroft(2)-PolohaPrijimace(2));
ChybyOdhaduBancroft0saZ(t,k) = norm(OdhadPolohyBancroft(3)-PolohaPrijimace(3));

[OdhadPolohyNewton,KovMatNewton] = Funkce_Newton(PoziceDruzic,PSRsum,STDsumu,S0);
ChybyOdhaduNewton(t,k) = norm(OdhadPolohyNewton(1:3)-PolohaPrijimace);
ChybyOdhaduNewton0saX(t,k) = norm(OdhadPolohyNewton(1)-PolohaPrijimace(1));
ChybyOdhaduNewton0saY(t,k) = norm(OdhadPolohyNewton(2)-PolohaPrijimace(2));
ChybyOdhaduNewton0saZ(t,k) = norm(OdhadPolohyNewton(3)-PolohaPrijimace(3));
HodnoceniKMstopaNewton(t,k) = sqrt(trace(KovMatNewton(1:3,1:3)));
HodnoceniKMdeterminantNewton(t,k) = det(KovMatNewton(1:3,1:3));
Elipsoid68Newton(t,k) = Funkce_Elipsoid(OdhadPolohyNewton(1:3),KovMatNewton(1:3,1:3),0.68,PolohaPrijimace);
Elipsoid95Newton(t,k) = Funkce_Elipsoid(OdhadPolohyNewton(1:3),KovMatNewton(1:3,1:3),0.95,PolohaPrijimace);

[X2,P2] = Funkce_KalmanR2(PoziceDruzic,PSRsum,STDsumu,T,X2,P2);
OdhadPolohyKalmanR2 = X2([1,3,5]);
KovMatKalmanR2 = P2([1,3,5],[1,3,5]);
ChybyOdhaduKalmanR2(t,k) = norm(OdhadPolohyKalmanR2(1:3)-PolohaPrijimace);
ChybyOdhaduKalmanR20saX(t,k) = norm(OdhadPolohyKalmanR2(1)-PolohaPrijimace(1));
ChybyOdhaduKalmanR20saY(t,k) = norm(OdhadPolohyKalmanR2(2)-PolohaPrijimace(2));
ChybyOdhaduKalmanR20saZ(t,k) = norm(OdhadPolohyKalmanR2(3)-PolohaPrijimace(3));
HodnoceniKMstopaKalmanR2(t,k) = sqrt(trace(KovMatKalmanR2(1:3,1:3)));
HodnoceniKMdeterminantKalmanR2(t,k) = det(KovMatKalmanR2(1:3,1:3));
Elipsoid68KalmanR2(t,k) = Funkce_Elipsoid(OdhadPolohyKalmanR2(1:3),KovMatKalmanR2(1:3,1:3),0.68,PolohaPrijimace);
Elipsoid95KalmanR2(t,k) = Funkce_Elipsoid(OdhadPolohyKalmanR2(1:3),KovMatKalmanR2(1:3,1:3),0.95,PolohaPrijimace);

[X3,P3] = Funkce_KalmanR3(PoziceDruzic,PSRsum,STDsumu,T,X3,P3);
OdhadPolohyKalmanR3 = X3([1,4,7]);
KovMatKalmanR3 = P3([1,4,7],[1,4,7]);
ChybyOdhaduKalmanR3(t,k) = norm(OdhadPolohyKalmanR3(1:3)-PolohaPrijimace);
ChybyOdhaduKalmanR30saX(t,k) = norm(OdhadPolohyKalmanR3(1)-PolohaPrijimace(1));
ChybyOdhaduKalmanR30saY(t,k) = norm(OdhadPolohyKalmanR3(2)-PolohaPrijimace(2));
ChybyOdhaduKalmanR30saZ(t,k) = norm(OdhadPolohyKalmanR3(3)-PolohaPrijimace(3));

```

```

% Vyvolání funkce pro výpočet PDOP pro jednotl. čas. okam.
% Přiřazení PSR k jednotl. družicím a čas. okam.
% Přičtení časové diference GPST a hodin družic k PSR
% Přiřazení šumu k PSR
% Přiřazení označení družic k jednotl. čas. okam.
% Nalezení družice, jež bude vysílat s poruchou

```

```

% Podmínka, pokud se jedná o daný čas. okam.
% Přičetní poruchy k dané družici
% Konec podmínky

```

```

% Vyvolání funkce pro výpočet polohy přijímače - Bancroft
% Vypočtení chyby odhadu - Bancroft
% Vypočtení chyby odhadu v ose X - Bancroft
% Vypočtení chyby odhadu v ose Y - Bancroft
% Vypočtení chyby odhadu v ose Z - Bancroft

```

```

% Vyvolání funkce pro výpočet polohy přijímače - Newton
% Vypočtení chyby odhadu - Newton
% Vypočtení chyby odhadu v ose X - Newton
% Vypočtení chyby odhadu v ose Y - Newton
% Vypočtení chyby odhadu v ose Z - Newton
% Vypočtení stopy KM - Newton
% Vypočtení determinantu KM - Newton
% Vyhodnocení, zdali je správný odhad - Newton (PST 68 %)
% Vyhodnocení, zdali je správný odhad - Newton (PST 95 %)

```

```

% Vyvolání funkce pro výpočet polohy přijímače - KalmanR2
% Přiřazení X z filtrace do proměnné OdhadPolohyKalmanR2
% Přiřazení P z filtrace do proměnné KovMatKalmanR2
% Vypočtení chyby odhadu - KalmanR2
% Vypočtení chyby odhadu v ose X - KalmanR2
% Vypočtení chyby odhadu v ose Y - KalmanR2
% Vypočtení chyby odhadu v ose Z - KalmanR2
% Vypočtení stopy KM - KalmanR2
% Vypočtení determinantu KM - KalmanR2
% Vyhodnocení, zdali je správný odhad - KalmanR2 (PST 68 %)
% Vyhodnocení, zdali je správný odhad - KalmanR2 (PST 95 %)

```

```

% Vyvolání funkce pro výpočet polohy přijímače - KalmanR3
% Přiřazení X z filtrace do proměnné OdhadPolohyKalmanR3
% Přiřazení P z filtrace do proměnné KovMatKalmanR3
% Vypočtení chyby odhadu - KalmanR3
% Vypočtení chyby odhadu v ose X - KalmanR3
% Vypočtení chyby odhadu v ose Y - KalmanR3
% Vypočtení chyby odhadu v ose Z - KalmanR3

```



```

HodnoceniKMstopaKalmanR3(t,k) = sqrt(trace(KovMatKalmanR3(1:3,1:3))); % Vypočtení stopy KM - KalmanR3
HodnoceniKMdeterminantKalmanR3(t,k) = det(KovMatKalmanR3(1:3,1:3)); % Vypočtení determinantu KM - KalmanR3
Elipsoid68KalmanR3(t,k) = Funkce_Elipsoid(OdhadPolohyKalmanR3(1:3),KovMatKalmanR3(1:3,1:3),0.68,PolohaPrijimace); % Vyhodnocení, zdali je správný odhad - KalmanR3 (PST 68 %)
Elipsoid95KalmanR3(t,k) = Funkce_Elipsoid(OdhadPolohyKalmanR3(1:3),KovMatKalmanR3(1:3,1:3),0.95,PolohaPrijimace); % Vyhodnocení, zdali je správný odhad - KalmanR3 (PST 95 %)
end % Konec vnitřního cyklu FOR

Den = k % Zobrazení pořadí dne
toc % Zobrazení uplynulého času
end % Konec cyklu FOR

```

## VYHODNOCENÍ SIMULACE

```

% EXPORT DO MS EXCEL
xlswrite('CasOkamPDOP.xlsx',CasOkamPDOP);
xlswrite('ChybyOdhaduBancroft.xlsx',ChybyOdhaduBancroft);
xlswrite('ChybyOdhaduBancroft0saX.xlsx',ChybyOdhaduBancroft0saX);
xlswrite('ChybyOdhaduBancroft0saY.xlsx',ChybyOdhaduBancroft0saY);
xlswrite('ChybyOdhaduBancroft0saZ.xlsx',ChybyOdhaduBancroft0saZ);
xlswrite('ChybyOdhaduNewton.xlsx',ChybyOdhaduNewton);
xlswrite('ChybyOdhaduNewton0saX.xlsx',ChybyOdhaduNewton0saX);
xlswrite('ChybyOdhaduNewton0saY.xlsx',ChybyOdhaduNewton0saY);
xlswrite('ChybyOdhaduNewton0saZ.xlsx',ChybyOdhaduNewton0saZ);
xlswrite('ChybyOdhaduKalmanR2.xlsx',ChybyOdhaduKalmanR2);
xlswrite('ChybyOdhaduKalmanR20saX.xlsx',ChybyOdhaduKalmanR20saX);
xlswrite('ChybyOdhaduKalmanR20saY.xlsx',ChybyOdhaduKalmanR20saY);
xlswrite('ChybyOdhaduKalmanR20saZ.xlsx',ChybyOdhaduKalmanR20saZ);
xlswrite('ChybyOdhaduKalmanR3.xlsx',ChybyOdhaduKalmanR3);
xlswrite('ChybyOdhaduKalmanR30saX.xlsx',ChybyOdhaduKalmanR30saX);
xlswrite('ChybyOdhaduKalmanR30saY.xlsx',ChybyOdhaduKalmanR30saY);
xlswrite('ChybyOdhaduKalmanR30saZ.xlsx',ChybyOdhaduKalmanR30saZ);
xlswrite('HodnoceniKMstopaNewton.xlsx',HodnoceniKMstopaNewton);
xlswrite('HodnoceniKMdeterminantNewton.xlsx',HodnoceniKMdeterminantNewton);
xlswrite('HodnoceniKMstopaKalmanR2.xlsx',HodnoceniKMstopaKalmanR2);
xlswrite('HodnoceniKMdeterminantKalmanR2.xlsx',HodnoceniKMdeterminantKalmanR2);
xlswrite('HodnoceniKMstopaKalmanR3.xlsx',HodnoceniKMstopaKalmanR3);
xlswrite('HodnoceniKMdeterminantKalmanR3.xlsx',HodnoceniKMdeterminantKalmanR3);
xlswrite('Elipsoid68Newton.xlsx',Elipsoid68Newton);
xlswrite('Elipsoid95Newton.xlsx',Elipsoid95Newton);
xlswrite('Elipsoid68KalmanR2.xlsx',Elipsoid68KalmanR2);
xlswrite('Elipsoid95KalmanR2.xlsx',Elipsoid95KalmanR2);
xlswrite('Elipsoid68KalmanR3.xlsx',Elipsoid68KalmanR3);
xlswrite('Elipsoid95KalmanR3.xlsx',Elipsoid95KalmanR3);
xlswrite('DateStart.xlsx',DateStart);
xlswrite('STDsumu.xlsx',STDsumu);

```

```
% KLASIFIKACE DLE PDOP
PDOP12 = (CasOkamPDOP > 1)+(CasOkamPDOP <= 2) == 2;
PDOP23 = (CasOkamPDOP > 2)+(CasOkamPDOP <= 3) == 2;
PDOP34 = (CasOkamPDOP > 3)+(CasOkamPDOP <= 4) == 2;
PDOP45 = (CasOkamPDOP > 4)+(CasOkamPDOP <= 5) == 2;
PDOP56 = (CasOkamPDOP > 5)+(CasOkamPDOP <= 6) == 2;
PDOP610 = (CasOkamPDOP > 6)+(CasOkamPDOP <= 10) == 2;
PDOP1020 = (CasOkamPDOP > 10)+(CasOkamPDOP <= 20) == 2;
PDOP2055 = (CasOkamPDOP > 20)+(CasOkamPDOP <= 55) == 2;

% MIN, MEAN, MAX HODNOTY CHYB ODHADŮ - BANCROFT
ChybyOdhaduBancroftPDOP12 = ChybyOdhaduBancroft(PDOP12,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP12 = max(ChybyOdhaduBancroftPDOP12(:));
MinChybaOdhaduBancroftPDOP12 = min(ChybyOdhaduBancroftPDOP12(:));
MeanChybaOdhaduBancroftPDOP12 = mean(ChybyOdhaduBancroftPDOP12(:));
ChybyOdhaduBancroftPDOP23 = ChybyOdhaduBancroft(PDOP23,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP23 = max(ChybyOdhaduBancroftPDOP23(:));
MinChybaOdhaduBancroftPDOP23 = min(ChybyOdhaduBancroftPDOP23(:));
MeanChybaOdhaduBancroftPDOP23 = mean(ChybyOdhaduBancroftPDOP23(:));
ChybyOdhaduBancroftPDOP34 = ChybyOdhaduBancroft(PDOP34,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP34 = max(ChybyOdhaduBancroftPDOP34(:));
MinChybaOdhaduBancroftPDOP34 = min(ChybyOdhaduBancroftPDOP34(:));
MeanChybaOdhaduBancroftPDOP34 = mean(ChybyOdhaduBancroftPDOP34(:));
ChybyOdhaduBancroftPDOP45 = ChybyOdhaduBancroft(PDOP45,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP45 = max(ChybyOdhaduBancroftPDOP45(:));
MinChybaOdhaduBancroftPDOP45 = min(ChybyOdhaduBancroftPDOP45(:));
MeanChybaOdhaduBancroftPDOP45 = mean(ChybyOdhaduBancroftPDOP45(:));
ChybyOdhaduBancroftPDOP56 = ChybyOdhaduBancroft(PDOP56,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP56 = max(ChybyOdhaduBancroftPDOP56(:));
MinChybaOdhaduBancroftPDOP56 = min(ChybyOdhaduBancroftPDOP56(:));
MeanChybaOdhaduBancroftPDOP56 = mean(ChybyOdhaduBancroftPDOP56(:));
ChybyOdhaduBancroftPDOP610 = ChybyOdhaduBancroft(PDOP610,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP610 = max(ChybyOdhaduBancroftPDOP610(:));
MinChybaOdhaduBancroftPDOP610 = min(ChybyOdhaduBancroftPDOP610(:));
MeanChybaOdhaduBancroftPDOP610 = mean(ChybyOdhaduBancroftPDOP610(:));
ChybyOdhaduBancroftPDOP1020 = ChybyOdhaduBancroft(PDOP1020,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP1020 = max(ChybyOdhaduBancroftPDOP1020(:));
MinChybaOdhaduBancroftPDOP1020 = min(ChybyOdhaduBancroftPDOP1020(:));
MeanChybaOdhaduBancroftPDOP1020 = mean(ChybyOdhaduBancroftPDOP1020(:));
ChybyOdhaduBancroftPDOP2055 = ChybyOdhaduBancroft(PDOP2055,1:PocetDnu);
MaxChybaOdhaduBancroftPDOP2055 = max(ChybyOdhaduBancroftPDOP2055(:));
MinChybaOdhaduBancroftPDOP2055 = min(ChybyOdhaduBancroftPDOP2055(:));
MeanChybaOdhaduBancroftPDOP2055 = mean(ChybyOdhaduBancroftPDOP2055(:));
```

```
% MIN, MEAN, MAX HODNOTY CHYB ODHADŮ - NEWTON
```

```
ChybyOdhaduNewtonPDOP12 = ChybyOdhaduNewton(PDOP12,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP12 = max(ChybyOdhaduNewtonPDOP12(:));  
MinChybaOdhaduNewtonPDOP12 = min(ChybyOdhaduNewtonPDOP12(:));  
MeanChybaOdhaduNewtonPDOP12 = mean(ChybyOdhaduNewtonPDOP12(:));  
ChybyOdhaduNewtonPDOP23 = ChybyOdhaduNewton(PDOP23,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP23 = max(ChybyOdhaduNewtonPDOP23(:));  
MinChybaOdhaduNewtonPDOP23 = min(ChybyOdhaduNewtonPDOP23(:));  
MeanChybaOdhaduNewtonPDOP23 = mean(ChybyOdhaduNewtonPDOP23(:));  
ChybyOdhaduNewtonPDOP34 = ChybyOdhaduNewton(PDOP34,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP34 = max(ChybyOdhaduNewtonPDOP34(:));  
MinChybaOdhaduNewtonPDOP34 = min(ChybyOdhaduNewtonPDOP34(:));  
MeanChybaOdhaduNewtonPDOP34 = mean(ChybyOdhaduNewtonPDOP34(:));  
ChybyOdhaduNewtonPDOP45 = ChybyOdhaduNewton(PDOP45,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP45 = max(ChybyOdhaduNewtonPDOP45(:));  
MinChybaOdhaduNewtonPDOP45 = min(ChybyOdhaduNewtonPDOP45(:));  
MeanChybaOdhaduNewtonPDOP45 = mean(ChybyOdhaduNewtonPDOP45(:));  
ChybyOdhaduNewtonPDOP56 = ChybyOdhaduNewton(PDOP56,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP56 = max(ChybyOdhaduNewtonPDOP56(:));  
MinChybaOdhaduNewtonPDOP56 = min(ChybyOdhaduNewtonPDOP56(:));  
MeanChybaOdhaduNewtonPDOP56 = mean(ChybyOdhaduNewtonPDOP56(:));  
ChybyOdhaduNewtonPDOP610 = ChybyOdhaduNewton(PDOP610,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP610 = max(ChybyOdhaduNewtonPDOP610(:));  
MinChybaOdhaduNewtonPDOP610 = min(ChybyOdhaduNewtonPDOP610(:));  
MeanChybaOdhaduNewtonPDOP610 = mean(ChybyOdhaduNewtonPDOP610(:));  
ChybyOdhaduNewtonPDOP1020 = ChybyOdhaduNewton(PDOP1020,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP1020 = max(ChybyOdhaduNewtonPDOP1020(:));  
MinChybaOdhaduNewtonPDOP1020 = min(ChybyOdhaduNewtonPDOP1020(:));  
MeanChybaOdhaduNewtonPDOP1020 = mean(ChybyOdhaduNewtonPDOP1020(:));  
ChybyOdhaduNewtonPDOP2055 = ChybyOdhaduNewton(PDOP2055,1:PocetDnu);  
MaxChybaOdhaduNewtonPDOP2055 = max(ChybyOdhaduNewtonPDOP2055(:));  
MinChybaOdhaduNewtonPDOP2055 = min(ChybyOdhaduNewtonPDOP2055(:));  
MeanChybaOdhaduNewtonPDOP2055 = mean(ChybyOdhaduNewtonPDOP2055(:));
```

```
% MIN, MEAN, MAX HODNOTY CHYB ODHADŮ - KALMAN R2
```

```
ChybyOdhaduKalmanR2PDOP12 = ChybyOdhaduKalmanR2(PDOP12,1:PocetDnu);  
MaxChybaOdhaduKalmanR2PDOP12 = max(ChybyOdhaduKalmanR2PDOP12(:));  
MinChybaOdhaduKalmanR2PDOP12 = min(ChybyOdhaduKalmanR2PDOP12(:));  
MeanChybaOdhaduKalmanR2PDOP12 = mean(ChybyOdhaduKalmanR2PDOP12(:));  
ChybyOdhaduKalmanR2PDOP23 = ChybyOdhaduKalmanR2(PDOP23,1:PocetDnu);  
MaxChybaOdhaduKalmanR2PDOP23 = max(ChybyOdhaduKalmanR2PDOP23(:));  
MinChybaOdhaduKalmanR2PDOP23 = min(ChybyOdhaduKalmanR2PDOP23(:));  
MeanChybaOdhaduKalmanR2PDOP23 = mean(ChybyOdhaduKalmanR2PDOP23(:));  
ChybyOdhaduKalmanR2PDOP34 = ChybyOdhaduKalmanR2(PDOP34,1:PocetDnu);  
MaxChybaOdhaduKalmanR2PDOP34 = max(ChybyOdhaduKalmanR2PDOP34(:));
```

```
MinChybaOdhaduKalmanR2PDOP34 = min(ChybyOdhaduKalmanR2PDOP34(:));
MeanChybaOdhaduKalmanR2PDOP34 = mean(ChybyOdhaduKalmanR2PDOP34(:));
ChybyOdhaduKalmanR2PDOP45 = ChybyOdhaduKalmanR2(PDOP45,1:PocetDnu);
MaxChybaOdhaduKalmanR2PDOP45 = max(ChybyOdhaduKalmanR2PDOP45(:));
MinChybaOdhaduKalmanR2PDOP45 = min(ChybyOdhaduKalmanR2PDOP45(:));
MeanChybaOdhaduKalmanR2PDOP45 = mean(ChybyOdhaduKalmanR2PDOP45(:));
ChybyOdhaduKalmanR2PDOP56 = ChybyOdhaduKalmanR2(PDOP56,1:PocetDnu);
MaxChybaOdhaduKalmanR2PDOP56 = max(ChybyOdhaduKalmanR2PDOP56(:));
MinChybaOdhaduKalmanR2PDOP56 = min(ChybyOdhaduKalmanR2PDOP56(:));
MeanChybaOdhaduKalmanR2PDOP56 = mean(ChybyOdhaduKalmanR2PDOP56(:));
ChybyOdhaduKalmanR2PDOP610 = ChybyOdhaduKalmanR2(PDOP610,1:PocetDnu);
MaxChybaOdhaduKalmanR2PDOP610 = max(ChybyOdhaduKalmanR2PDOP610(:));
MinChybaOdhaduKalmanR2PDOP610 = min(ChybyOdhaduKalmanR2PDOP610(:));
MeanChybaOdhaduKalmanR2PDOP610 = mean(ChybyOdhaduKalmanR2PDOP610(:));
ChybyOdhaduKalmanR2PDOP1020 = ChybyOdhaduKalmanR2(PDOP1020,1:PocetDnu);
MaxChybaOdhaduKalmanR2PDOP1020 = max(ChybyOdhaduKalmanR2PDOP1020(:));
MinChybaOdhaduKalmanR2PDOP1020 = min(ChybyOdhaduKalmanR2PDOP1020(:));
MeanChybaOdhaduKalmanR2PDOP1020 = mean(ChybyOdhaduKalmanR2PDOP1020(:));
ChybyOdhaduKalmanR2PDOP2055 = ChybyOdhaduKalmanR2(PDOP2055,1:PocetDnu);
MaxChybaOdhaduKalmanR2PDOP2055 = max(ChybyOdhaduKalmanR2PDOP2055(:));
MinChybaOdhaduKalmanR2PDOP2055 = min(ChybyOdhaduKalmanR2PDOP2055(:));
MeanChybaOdhaduKalmanR2PDOP2055 = mean(ChybyOdhaduKalmanR2PDOP2055(:));
```

% MIN, MEAN, MAX HODNOTY CHYB ODHADŮ - KALMAN R3

```
ChybyOdhaduKalmanR3PDOP12 = ChybyOdhaduKalmanR3(PDOP12,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP12 = max(ChybyOdhaduKalmanR3PDOP12(:));
MinChybaOdhaduKalmanR3PDOP12 = min(ChybyOdhaduKalmanR3PDOP12(:));
MeanChybaOdhaduKalmanR3PDOP12 = mean(ChybyOdhaduKalmanR3PDOP12(:));
ChybyOdhaduKalmanR3PDOP23 = ChybyOdhaduKalmanR3(PDOP23,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP23 = max(ChybyOdhaduKalmanR3PDOP23(:));
MinChybaOdhaduKalmanR3PDOP23 = min(ChybyOdhaduKalmanR3PDOP23(:));
MeanChybaOdhaduKalmanR3PDOP23 = mean(ChybyOdhaduKalmanR3PDOP23(:));
ChybyOdhaduKalmanR3PDOP34 = ChybyOdhaduKalmanR3(PDOP34,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP34 = max(ChybyOdhaduKalmanR3PDOP34(:));
MinChybaOdhaduKalmanR3PDOP34 = min(ChybyOdhaduKalmanR3PDOP34(:));
MeanChybaOdhaduKalmanR3PDOP34 = mean(ChybyOdhaduKalmanR3PDOP34(:));
ChybyOdhaduKalmanR3PDOP45 = ChybyOdhaduKalmanR3(PDOP45,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP45 = max(ChybyOdhaduKalmanR3PDOP45(:));
MinChybaOdhaduKalmanR3PDOP45 = min(ChybyOdhaduKalmanR3PDOP45(:));
MeanChybaOdhaduKalmanR3PDOP45 = mean(ChybyOdhaduKalmanR3PDOP45(:));
ChybyOdhaduKalmanR3PDOP56 = ChybyOdhaduKalmanR3(PDOP56,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP56 = max(ChybyOdhaduKalmanR3PDOP56(:));
MinChybaOdhaduKalmanR3PDOP56 = min(ChybyOdhaduKalmanR3PDOP56(:));
MeanChybaOdhaduKalmanR3PDOP56 = mean(ChybyOdhaduKalmanR3PDOP56(:));
ChybyOdhaduKalmanR3PDOP610 = ChybyOdhaduKalmanR3(PDOP610,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP610 = max(ChybyOdhaduKalmanR3PDOP610(:));
```

```
MinChybaOdhaduKalmanR3PDOP610 = min(ChybyOdhaduKalmanR3PDOP610(:));
MeanChybaOdhaduKalmanR3PDOP610 = mean(ChybyOdhaduKalmanR3PDOP610(:));
ChybyOdhaduKalmanR3PDOP1020 = ChybyOdhaduKalmanR3(PDOP1020,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP1020 = max(ChybyOdhaduKalmanR3PDOP1020(:));
MinChybaOdhaduKalmanR3PDOP1020 = min(ChybyOdhaduKalmanR3PDOP1020(:));
MeanChybaOdhaduKalmanR3PDOP1020 = mean(ChybyOdhaduKalmanR3PDOP1020(:));
ChybyOdhaduKalmanR3PDOP2055 = ChybyOdhaduKalmanR3(PDOP2055,1:PocetDnu);
MaxChybaOdhaduKalmanR3PDOP2055 = max(ChybyOdhaduKalmanR3PDOP2055(:));
MinChybaOdhaduKalmanR3PDOP2055 = min(ChybyOdhaduKalmanR3PDOP2055(:));
MeanChybaOdhaduKalmanR3PDOP2055 = mean(ChybyOdhaduKalmanR3PDOP2055(:));

% PROCENTUÁLNÍ NÁLEŽITOST DO ELIPSOIDU
Elipsoid68NewtonSum = sum(Elipsoid68Newton,2);
Elipsoid68KalmanR2Sum = sum(Elipsoid68KalmanR2,2);
Elipsoid68KalmanR3Sum = sum(Elipsoid68KalmanR3,2);
Elipsoid95NewtonSum = sum(Elipsoid95Newton,2);
Elipsoid95KalmanR2Sum = sum(Elipsoid95KalmanR2,2);
Elipsoid95KalmanR3Sum = sum(Elipsoid95KalmanR3,2);
Elipsoid68NewtonProcentNa1 = (Elipsoid68NewtonSum/PocetDnu)*100;
Elipsoid68KalmanR2ProcentNa1 = (Elipsoid68KalmanR2Sum/PocetDnu)*100;
Elipsoid68KalmanR3ProcentNa1 = (Elipsoid68KalmanR3Sum/PocetDnu)*100;
Elipsoid95NewtonProcentNa1 = (Elipsoid95NewtonSum/PocetDnu)*100;
Elipsoid95KalmanR2ProcentNa1 = (Elipsoid95KalmanR2Sum/PocetDnu)*100;
Elipsoid95KalmanR3ProcentNa1 = (Elipsoid95KalmanR3Sum/PocetDnu)*100;

% ROZDÍL ODMOCNIN STOP KM
RozdilKMstopaNewtonKalmanR2 = abs(HodnoceniKMstopaNewton(:,1)-HodnoceniKMstopaKalmanR2(:,1));
RozdilKMstopaNewtonKalmanR3 = abs(HodnoceniKMstopaNewton(:,1)-HodnoceniKMstopaKalmanR3(:,1));
RozdilKMstopaKalmanR2KalmanR3 = abs(HodnoceniKMstopaKalmanR2(:,1)-HodnoceniKMstopaKalmanR3(:,1));

% ROZDÍL DETERMINANTŮ KM
RozdilKMdeterminantNewtonKalmanR2 = abs(HodnoceniKMdeterminantNewton(:,1)-HodnoceniKMdeterminantKalmanR2(:,1));
RozdilKMdeterminantNewtonKalmanR3 = abs(HodnoceniKMdeterminantNewton(:,1)-HodnoceniKMdeterminantKalmanR3(:,1));
RozdilKMdeterminantKalmanR2KalmanR3 = abs(HodnoceniKMdeterminantKalmanR2(:,1)-HodnoceniKMdeterminantKalmanR3(:,1));

% GRAF 1 - HISTOGRAM PRO PDOP / PDOP DLE ČAS. OKAM.
figure(1)
subplot(2,1,1);
histogram(CasOkamPDOP(1:PocetOkamziku,1), 'FaceColor', [0.55,0,0.55]);
grid on
axis([0,10,0,2000]);
title('HISTOGRAM PRO HODNOTY PDOP');
xlabel('PDOP');
ylabel('Četnost');
subplot(2,1,2);
plot(CasOkamPDOP(1:PocetOkamziku,1), '.', 'color', [0.6,0.8,0.2]);
```

```
grid on
axis([0,PocetOkamziku,0,20]);
title('HODNOTY PDOP');
xlabel('Časový okamžik');
ylabel('PDOP');
savefig('Figure1');
saveas('Figure1','pdf');

% GRAF 2 - CHYBY ODHADU DLE ČAS. OKAM. (VŠE V JEDNOM)
figure(2)
subplot(2,1,1);
plot(ChybyOdhaduBancroft(1:PocetOkamziku,1),'color',[0.55,0,0]);
grid on
hold on
plot(ChybyOdhaduNewton(1:PocetOkamziku,1),'color',[0.13,0.55,0.13]);
hold on
plot(ChybyOdhaduKalmanR2(1:PocetOkamziku,1),'color',[0.25,0.41,0.88]);
hold on
plot(ChybyOdhaduKalmanR3(1:PocetOkamziku,1),'color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
legend('Bancroft','Newton','Kalman R2','Kalman R3','Location','Eastoutside');
subplot(2,1,2);
plot(ChybyOdhaduBancroft(1:PocetOkamziku,1),'color',[0.55,0,0]);
grid on
hold on
plot(ChybyOdhaduNewton(1:PocetOkamziku,1),'color',[0.13,0.55,0.13]);
hold on
plot(ChybyOdhaduKalmanR2(1:PocetOkamziku,1),'color',[0.25,0.41,0.88]);
hold on
plot(ChybyOdhaduKalmanR3(1:PocetOkamziku,1),'color',[0.72,0.53,0.04]);
axis([4650,4800,0,50]);
title('CHYBY ODHADU POLOHY');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
legend('Bancroft','Newton','Kalman R2','Kalman R3','Location','Eastoutside');
savefig('Figure2');
saveas('Figure2','pdf');

% GRAF 3 - CHYBY ODHADU DLE ČAS. OKAM. (SEPARÁTNĚ)
figure(3)
subplot(4,1,1);
plot(ChybyOdhaduBancroft(1:PocetOkamziku,1),'color',[0.55,0,0]);
```

```
grid on
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY - BANCROFT');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
subplot(4,1,2);
plot(ChybyOdhaduNewton(1:PocetOkamziku,1),'color',[0.13,0.55,0.13]);
grid on
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY - NEWTON');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
subplot(4,1,3);
plot(ChybyOdhaduKalmanR2(1:PocetOkamziku,1),'color',[0.25,0.41,0.88]);
grid on
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY - KALMAN R2');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
subplot(4,1,4)
plot(ChybyOdhaduKalmanR3(1:PocetOkamziku,1),'color',[0.72,0.53,0.04]);
grid on
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY - KALMAN R3');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
savefig('figure(3)', 'Figure3');
saveas('figure(3)', 'Figure3', 'pdf');

% GRAF 4 - CHYBY ODAHDU V OSÁCH X,Y,Z DLE ČAS. OKAM.
figure(4)
subplot(3,1,1);
plot(ChybyOdhaduBancroft0saX(1:PocetOkamziku,1),'color',[0.55,0,0]);
grid on
hold on
plot(ChybyOdhaduNewton0saX(1:PocetOkamziku,1),'color',[0.13,0.55,0.13]);
hold on
plot(ChybyOdhaduKalmanR20saX(1:PocetOkamziku,1),'color',[0.25,0.41,0.88]);
hold on
plot(ChybyOdhaduKalmanR30saX(1:PocetOkamziku,1),'color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY VE SMĚRU X');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
legend('Bancroft', 'Newton', 'Kalman R2', 'Kalman R3', 'Location', 'Eastoutside');
```

```
subplot(3,1,2);
plot(ChybyOdhaduBancroft0saY(1:PocetOkamziku,1), 'color', [0.55,0,0]);
grid on
hold on
plot(ChybyOdhaduNewton0saY(1:PocetOkamziku,1), 'color', [0.13,0.55,0.13]);
hold on
plot(ChybyOdhaduKalmanR20saY(1:PocetOkamziku,1), 'color', [0.25,0.41,0.88]);
hold on
plot(ChybyOdhaduKalmanR30saY(1:PocetOkamziku,1), 'color', [0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY VE SMĚRU Y');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
legend('Bancroft', 'Newton', 'Kalman R2', 'Kalman R3', 'Location', 'Eastoutside');
subplot(3,1,3);
plot(ChybyOdhaduBancroft0saZ(1:PocetOkamziku,1), 'color', [0.55,0,0]);
grid on
hold on
plot(ChybyOdhaduNewton0saZ(1:PocetOkamziku,1), 'color', [0.13,0.55,0.13]);
hold on
plot(ChybyOdhaduKalmanR20saZ(1:PocetOkamziku,1), 'color', [0.25,0.41,0.88]);
hold on
plot(ChybyOdhaduKalmanR30saZ(1:PocetOkamziku,1), 'color', [0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,50]);
title('CHYBY ODHADU POLOHY VE SMĚRU Z');
xlabel('Časový okamžik');
ylabel('Chyba odhadu [m]');
legend('Bancroft', 'Newton', 'Kalman R2', 'Kalman R3', 'Location', 'Eastoutside');
savefig('figure(4)', 'Figure4');
saveas('figure(4)', 'Figure4', 'pdf');

% GRAF 5 - MIN, MEAN, MAX HODNOTY CHYB ODHADU DLE PDOP
figure(5)
subplot(3,1,1);
plot(1,MaxChybaOdhaduBancroftPDOP12, 'color', [0.55,0,0], 'marker', '.', 'markersize', 20);
grid on
hold on
plot(1,MaxChybaOdhaduNewtonPDOP12, 'color', [0.13,0.55,0.13], 'marker', '.', 'markersize', 20);
hold on
plot(1,MaxChybaOdhaduKalmanR2PDOP12, 'color', [0.25,0.41,0.88], 'marker', '.', 'markersize', 20);
hold on
plot(1,MaxChybaOdhaduKalmanR3PDOP12, 'color', [0.72,0.53,0.04], 'marker', '.', 'markersize', 20);
hold on
plot(2,MaxChybaOdhaduBancroftPDOP23, 'color', [0.55,0,0], 'marker', '.', 'markersize', 20);
hold on
```



```
plot(2,MaxChybaOdhaduNewtonPDOP23,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(2,MaxChybaOdhaduKalmanR2PDOP23,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(2,MaxChybaOdhaduKalmanR3PDOP23,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(3,MaxChybaOdhaduBancroftPDOP34,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(3,MaxChybaOdhaduNewtonPDOP34,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(3,MaxChybaOdhaduKalmanR2PDOP34,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(3,MaxChybaOdhaduKalmanR3PDOP34,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(4,MaxChybaOdhaduBancroftPDOP45,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(4,MaxChybaOdhaduNewtonPDOP45,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(4,MaxChybaOdhaduKalmanR2PDOP45,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(4,MaxChybaOdhaduKalmanR3PDOP45,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(5,MaxChybaOdhaduBancroftPDOP56,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(5,MaxChybaOdhaduNewtonPDOP56,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(5,MaxChybaOdhaduKalmanR2PDOP56,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(5,MaxChybaOdhaduKalmanR3PDOP56,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(6,MaxChybaOdhaduBancroftPDOP610,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(6,MaxChybaOdhaduNewtonPDOP610,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(6,MaxChybaOdhaduKalmanR2PDOP610,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(6,MaxChybaOdhaduKalmanR3PDOP610,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(7,MaxChybaOdhaduBancroftPDOP1020,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(7,MaxChybaOdhaduNewtonPDOP1020,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(7,MaxChybaOdhaduKalmanR2PDOP1020,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(7,MaxChybaOdhaduKalmanR3PDOP1020,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
```

```
hold on
plot(8,MaxChybaOdhaduBancroftPDOP2055,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(8,MaxChybaOdhaduNewtonPDOP2055,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(8,MaxChybaOdhaduKalmanR2PDOP2055,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(8,MaxChybaOdhaduKalmanR3PDOP2055,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
axis([0,9,0,100]);
title('MAXIMÁLNÍ HODNOTY CHYB ODHADU POLOHY');
xlabel('PDOP');
xticklabels({'0','1 - 2','2 - 3','3 - 4','4 - 5','5 - 6','6 - 10','10 - 20','20 - 55'});
ylabel('Maximální hodnota chyby odhadu [m]');
legend('Bancroft','Newton','Kalman R2','Kalman R3','Location','Eastoutside');
subplot(3,1,2);
plot(1,MeanChybaOdhaduBancroftPDOP12,'color',[0.55,0,0],'marker','.', 'markersize',20);
grid on
hold on
plot(1,MeanChybaOdhaduNewtonPDOP12,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(1,MeanChybaOdhaduKalmanR2PDOP12,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(1,MeanChybaOdhaduKalmanR3PDOP12,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(2,MeanChybaOdhaduBancroftPDOP23,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(2,MeanChybaOdhaduNewtonPDOP23,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(2,MeanChybaOdhaduKalmanR2PDOP23,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(2,MeanChybaOdhaduKalmanR3PDOP23,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(3,MeanChybaOdhaduBancroftPDOP34,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(3,MeanChybaOdhaduNewtonPDOP34,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(3,MeanChybaOdhaduKalmanR2PDOP34,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(3,MeanChybaOdhaduKalmanR3PDOP34,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(4,MeanChybaOdhaduBancroftPDOP45,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(4,MeanChybaOdhaduNewtonPDOP45,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(4,MeanChybaOdhaduKalmanR2PDOP45,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
```

```
hold on
plot(4,MeanChybaOdhaduKalmanR3PDOP45,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(5,MeanChybaOdhaduBancroftPDOP56,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(5,MeanChybaOdhaduNewtonPDOP56,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(5,MeanChybaOdhaduKalmanR2PDOP56,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(5,MeanChybaOdhaduKalmanR3PDOP56,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(6,MeanChybaOdhaduBancroftPDOP610,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(6,MeanChybaOdhaduNewtonPDOP610,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(6,MeanChybaOdhaduKalmanR2PDOP610,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(6,MeanChybaOdhaduKalmanR3PDOP610,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(7,MeanChybaOdhaduBancroftPDOP1020,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(7,MeanChybaOdhaduNewtonPDOP1020,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(7,MeanChybaOdhaduKalmanR2PDOP1020,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(7,MeanChybaOdhaduKalmanR3PDOP1020,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(8,MeanChybaOdhaduBancroftPDOP2055,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(8,MeanChybaOdhaduNewtonPDOP2055,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(8,MeanChybaOdhaduKalmanR2PDOP2055,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(8,MeanChybaOdhaduKalmanR3PDOP2055,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
axis([0,9,0,50]);
title('STŘEDNÍ HODNOTY CHYB ODHADU POLOHY');
xlabel('PDOP');
xtickLabels({'0','1 - 2','2 - 3','3 - 4','4 - 5','5 - 6','6 - 10','10 - 20','20 - 55'});
ylabel('Střední hodnota chyby odhadu [m]');
legend('Bancroft','Newton','Kalman R2','Kalman R3','Location','Eastoutside');
subplot(3,1,3);
plot(1,MinChybaOdhaduBancroftPDOP12,'color',[0.55,0,0],'marker','.', 'markersize',20);
grid on
hold on
plot(1,MinChybaOdhaduNewtonPDOP12,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
```

```
hold on
plot(1,MinChybaOdhaduKalmanR2PDOP12,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(1,MinChybaOdhaduKalmanR3PDOP12,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(2,MinChybaOdhaduBancroftPDOP23,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(2,MinChybaOdhaduNewtonPDOP23,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(2,MinChybaOdhaduKalmanR2PDOP23,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(2,MinChybaOdhaduKalmanR3PDOP23,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(3,MinChybaOdhaduBancroftPDOP34,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(3,MinChybaOdhaduNewtonPDOP34,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(3,MinChybaOdhaduKalmanR2PDOP34,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(3,MinChybaOdhaduKalmanR3PDOP34,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(4,MinChybaOdhaduBancroftPDOP45,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(4,MinChybaOdhaduNewtonPDOP45,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(4,MinChybaOdhaduKalmanR2PDOP45,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(4,MinChybaOdhaduKalmanR3PDOP45,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(5,MinChybaOdhaduBancroftPDOP56,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(5,MinChybaOdhaduNewtonPDOP56,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(5,MinChybaOdhaduKalmanR2PDOP56,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(5,MinChybaOdhaduKalmanR3PDOP56,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(6,MinChybaOdhaduBancroftPDOP610,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(6,MinChybaOdhaduNewtonPDOP610,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(6,MinChybaOdhaduKalmanR2PDOP610,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(6,MinChybaOdhaduKalmanR3PDOP610,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(7,MinChybaOdhaduBancroftPDOP1020,'color',[0.55,0,0],'marker','.', 'markersize',20);
```

```
hold on
plot(7,MinChybaOdhaduNewtonPDOP1020,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(7,MinChybaOdhaduKalmanR2PDOP1020,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(7,MinChybaOdhaduKalmanR3PDOP1020,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
hold on
plot(8,MinChybaOdhaduBancroftPDOP2055,'color',[0.55,0,0],'marker','.', 'markersize',20);
hold on
plot(8,MinChybaOdhaduNewtonPDOP2055,'color',[0.13,0.55,0.13],'marker','.', 'markersize',20);
hold on
plot(8,MinChybaOdhaduKalmanR2PDOP2055,'color',[0.25,0.41,0.88],'marker','.', 'markersize',20);
hold on
plot(8,MinChybaOdhaduKalmanR3PDOP2055,'color',[0.72,0.53,0.04],'marker','.', 'markersize',20);
axis([0,9,0,5]);
title('MINIMÁLNÍ HODNOTY CHYB ODHADU POLOHY');
xlabel('PDOP');
xticklabels({'0', '1 - 2', '2 - 3', '3 - 4', '4 - 5', '5 - 6', '6 - 10', '10 - 20', '20 - 55'});
ylabel('Minimální hodnota chyby odhadu [m]');
legend('Bancroft', 'Newton', 'Kalman R2', 'Kalman R3', 'Location', 'Eastoutside');
savefig('figure(5)', 'Figure5');
saveas('figure(5)', 'Figure5', 'pdf');

% GRAF 6 - WGS 84 A DRÁHY DRUŽIC
figure(6)
a = 6378137;
b = 6356752.3142;
SVindex10 = find(KONSTELACE_Ivan.druzice(:,2) == 10);
SVindex12 = find(KONSTELACE_Ivan.druzice(:,2) == 12);
SVindex21 = find(KONSTELACE_Ivan.druzice(:,2) == 21);
SVindex25 = find(KONSTELACE_Ivan.druzice(:,2) == 25);
SVindex29 = find(KONSTELACE_Ivan.druzice(:,2) == 29);
SVindex31 = find(KONSTELACE_Ivan.druzice(:,2) == 31);
syms uv vu
fsurf(a*sin(uv)*cos(vu), a*sin(uv)*sin(vu), b*cos(uv), [0,pi,0,2*pi]);
colormap([0.75,0.75,0.75]);
grid on
rotate3d on
hold on
plot3(PolohaPrijimace(1),PolohaPrijimace(2),PolohaPrijimace(3),'color',[1,0,0],'marker','.', 'markersize',40);
hold on
plot3(KONSTELACE_Ivan.druzice(SVindex10,3),KONSTELACE_Ivan.druzice(SVindex10,4),KONSTELACE_Ivan.druzice(SVindex10,5),'.', 'markersize',20);
hold on
plot3(KONSTELACE_Ivan.druzice(SVindex12,3),KONSTELACE_Ivan.druzice(SVindex12,4),KONSTELACE_Ivan.druzice(SVindex12,5),'.', 'markersize',20);
hold on
```

```
plot3(KONSTELACE_Ivan.druzice(SVindex21,3),KONSTELACE_Ivan.druzice(SVindex21,4),KONSTELACE_Ivan.druzice(SVindex21,5),'.','markersize',20);
hold on
plot3(KONSTELACE_Ivan.druzice(SVindex25,3),KONSTELACE_Ivan.druzice(SVindex25,4),KONSTELACE_Ivan.druzice(SVindex25,5),'.','markersize',20);
hold on
plot3(KONSTELACE_Ivan.druzice(SVindex29,3),KONSTELACE_Ivan.druzice(SVindex29,4),KONSTELACE_Ivan.druzice(SVindex29,5),'.','markersize',20);
hold on
plot3(KONSTELACE_Ivan.druzice(SVindex31,3),KONSTELACE_Ivan.druzice(SVindex31,4),KONSTELACE_Ivan.druzice(SVindex31,5),'.','markersize',20);
view([PolohaPrijimace(1),PolohaPrijimace(2),PolohaPrijimace(3)]);
title('WGS 84 & DRÁHY DRUŽIC')
xlabel('Souřadnice X [m]');
ylabel('Souřadnice Y [m]');
zlabel('Souřadnice Z [m]');
legend('Elipsoid WGS 84','GPS přijímač','SV 10','SV 12','SV 21','SV 25','SV 29','SV 31','Location','Eastoutside');
savefig(figure(6),'Figure6');
saveas(figure(6),'Figure6','pdf');

% GRAF 7 - PROCENTUÁLNÍ NÁLEŽITOST DO ELIPSOIDU
figure(7)
subplot(3,2,1);
grid on
plot(Elipsoid68NewtonProcentNal(1:PocetOkamziku,1),'.','color',[0.13,0.55,0.13]);
axis([0,PocetOkamziku,0,100]);
title('ELIPSOID 68 % - NEWTON');
xlabel('Časový okamžik');
ylabel('Procentuální náležitost [%]');
subplot(3,2,3);
grid on
plot(Elipsoid68KalmanR2ProcentNal(1:PocetOkamziku,1),'.','color',[0.25,0.41,0.88]);
axis([0,PocetOkamziku,0,100]);
title('ELIPSOID 68 % - KALMAN R2');
xlabel('Časový okamžik');
ylabel('Procentuální náležitost [%]');
subplot(3,2,5);
grid on
plot(Elipsoid68KalmanR3ProcentNal(1:PocetOkamziku,1),'.','color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,100]);
title('ELIPSOID 68 % - KALMAN R3');
xlabel('Časový okamžik');
ylabel('Procentuální náležitost [%]');
subplot(3,2,2);
grid on
plot(Elipsoid95NewtonProcentNal(1:PocetOkamziku,1),'.','color',[0.13,0.55,0.13]);
axis([0,PocetOkamziku,0,100]);
title('ELIPSOID 95 % - NEWTON');
xlabel('Časový okamžik');
```

```
ylabel('Procentuální náležitost [%]');
subplot(3,2,4);
grid on
plot(Elipsoid95KalmanR2ProcentNal(1:PocetOkamziku,1),'.','color',[0.25,0.41,0.88]);
axis([0,PocetOkamziku,0,100]);
title('ELIPSOID 95 % - KALMAN R2');
xlabel('Časový okamžik');
ylabel('Procentuální náležitost [%]');
subplot(3,2,6);
grid on
plot(Elipsoid95KalmanR3ProcentNal(1:PocetOkamziku,1),'.','color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,100]);
title('ELIPSOID 95 % - KALMAN R3');
xlabel('Časový okamžik');
ylabel('Procentuální náležitost [%]');
savefig(figure(7),'Figure7');
saveas(figure(7),'Figure7','pdf');

% GRAF 8 - ODMOCNINY STOP A DETERMINANTY KM
figure(8)
subplot(2,1,1);
grid on
plot(HodnoceniKMstopaNewton(1:PocetOkamziku,1),'.','color',[0.13,0.55,0.13]);
hold on
plot(HodnoceniKMstopaKalmanR2(1:PocetOkamziku,1),'.','color',[0.25,0.41,0.88]);
hold on
plot(HodnoceniKMstopaKalmanR3(1:PocetOkamziku,1),'.','color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,50]);
title('ODMOCNINA STOPY KM');
xlabel('Časový okamžik');
ylabel('Odmocnina stopy KM [m2]');
legend('Newton','Kalman R2','Kalman R3','Location','Eastoutside');
subplot(2,1,2);
grid on
plot(HodnoceniKMDeterminantNewton(1:PocetOkamziku,1),'.','color',[0.13,0.55,0.13]);
hold on
plot(HodnoceniKMDeterminantKalmanR2(1:PocetOkamziku,1),'.','color',[0.25,0.41,0.88]);
hold on
plot(HodnoceniKMDeterminantKalmanR3(1:PocetOkamziku,1),'.','color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,200]);
title('DETERMINANT KM');
xlabel('Časový okamžik');
ylabel('Determinant KM [m2]');
legend('Newton','Kalman R2','Kalman R3','Location','Eastoutside');
savefig(figure(8),'Figure8');
saveas(figure(8),'Figure8','pdf');
```

```
% GRAF 9 - ROZDÍL ODMOCNIN STOP A DETERMINANTŮ KM
figure(9)
subplot(3,2,1);
grid on
plot(RozdilKMstopaNewtonKalmanR2(1:PocetOkamziku,1),'.','color',[0.13,0.55,0.13]);
axis([0,PocetOkamziku,0,50]);
title('ROZDÍL ODMOCNIN STOPY KM (NEWTON - KALMAN R2)');
xlabel('Časový okamžik');
ylabel('Rozdíl odmocnin stopy KM [m2]');
subplot(3,2,3);
grid on
plot(RozdilKMstopaNewtonKalmanR3(1:PocetOkamziku,1),'.','color',[0.25,0.41,0.88]);
axis([0,PocetOkamziku,0,50]);
title('ROZDÍL ODMOCNIN STOPY KM (NEWTON - KALMAN R3)');
xlabel('Časový okamžik');
ylabel('Rozdíl odmocnin stopy KM [m2]');
subplot(3,2,5);
grid on
plot(RozdilKMstopaKalmanR2KalmanR3(1:PocetOkamziku,1),'.','color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,50]);
title('ROZDÍL ODMOCNIN STOPY KM (KALMAN R2 - KALMAN R3)');
xlabel('Časový okamžik');
ylabel('Rozdíl odmocnin stopy KM [m2]');
subplot(3,2,2);
grid on
plot(RozdilKMdeterminantNewtonKalmanR2(1:PocetOkamziku,1),'.','color',[0.13,0.55,0.13]);
axis([0,PocetOkamziku,0,1000]);
title('ROZDÍL DETERMINANTŮ STOPY KM (NEWTON - KALMAN R2)');
xlabel('Časový okamžik');
ylabel('Rozdíl determinantů KM [m2]');
subplot(3,2,4);
grid on
plot(RozdilKMdeterminantNewtonKalmanR3(1:PocetOkamziku,1),'.','color',[0.25,0.41,0.88]);
axis([0,PocetOkamziku,0,1000]);
title('ROZDÍL DETERMINANTŮ STOPY KM (NEWTON - KALMAN R3)');
xlabel('Časový okamžik');
ylabel('Rozdíl determinantů KM [m2]');
subplot(3,2,6);
grid on
plot(RozdilKMdeterminantKalmanR2KalmanR3(1:PocetOkamziku,1),'.','color',[0.72,0.53,0.04]);
axis([0,PocetOkamziku,0,1000]);
title('ROZDÍL DETERMINANTŮ STOPY KM (KALMAN R2 - KALMAN R3)');
xlabel('Časový okamžik');
ylabel('Rozdíl determinantů KM [m2]');
```



```

savefig(figure(9), 'Figure9');
saveas(figure(9), 'Figure9', 'pdf');

toc
clock;
DateEnd = fix(clock)
xlswrite('DateEnd.xlsx', DateEnd);

% Konec časovače (do této chvíle je počítána doba simulace)
% Funkce pro zobrazení dne a času
% Zobrazení dne a času začátku simulace [RRRR MM DD hh mm ss]
% Export do MS Excel

% FUNKCE PDOP
% -----
% Funkce spočte hodnotu PDOP
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016

```

## FUNKCE\_PDOP

```

function [PDOP] = Funkce_PDOP(PD, PPX, PPY, PPZ)
[PocetDruzic, ~] = size(PD);
A = NaN*ones(PocetDruzic, 4);

for n = 1:PocetDruzic
    A(n,1) = (PPX-PD(n,1))/sqrt((PPX-PD(n,1))^2+(PPY-PD(n,2))^2+(PPZ-PD(n,3))^2);
    A(n,2) = (PPY-PD(n,2))/sqrt((PPX-PD(n,1))^2+(PPY-PD(n,2))^2+(PPZ-PD(n,3))^2);
    A(n,3) = (PPZ-PD(n,3))/sqrt((PPX-PD(n,1))^2+(PPY-PD(n,2))^2+(PPZ-PD(n,3))^2);
    A(n,4) = 1;
end

ATA = A'*A;

if rank(ATA) == 4
    D = inv(ATA);
    PDOP = sqrt(D(1,1)+D(2,2)+D(3,3));
else
    PDOP = NaN;
end

end

```

% Funkce pro výpočet PDOP (vstupem jsou pozice družic a poloha přijímače v daný čas. okam.)  
 % Počet řádků odpovídá počtu družic v daném čas. okam. (počet sloupců je vždy 3)  
 % Matice "A" s hodnotami NaN

% Cyklus FOR od 1 do počtu družic v daném čas. okam.  
 % Vypočtení "A" (matice směrových kosinů)

% Počet řádků "A" odpovídá počtu družic v daném čas. okam.  
 % Konec cyklu FOR

% Součin transponované matice "A" s původní maticí "A"

% Podmínka, zdali je hodnota matice "ATA" rovna 4 (zdali je čtvercová)  
 % Je-li hodnota "ATA" rovna 4, provede se inverze této matice  
 % Výpočet PDOP z matice "D" (odmocnina stopy matice, ale bez času)

% Pokud není hodnota "ATA" rovna 4, nelze udělat inverzi a výpočet PDOP neproběhne  
 % Konec podmínky

% Konec funkce

```
% FUNKCE BANCROFT
% -----
% Funkce spočte odhad polohy dle Bancroftovy
% algebraické metody
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016
```

### FUNKCE\_BANCROFT

```
function [OdhadPolohyBancroft] = Funkce_Bancroft(PD,PSRsum)
[PocetDruzic,~] = size(PD);
mink = @(x,y) x(1)*y(1)+x(2)*y(2)+x(3)*y(3)-x(4)*y(4);
B = [PD PSRsum];
e = ones(PocetDruzic,1);
a = NaN*ones(PocetDruzic,1);

for n = 1:PocetDruzic
    a(n,1) = mink(B(n,:),B(n,:))/2;
end

BTB = B'*B;

if rank(BTB) == 4
    Bplus = inv(BTB)*B';
else
    Bplus = Inf;
end

u = Bplus*e;
v = Bplus*a;
E = mink(u,u);
F = mink(u,v)-1;
G = mink(v,v);
D = (2*F)^2-4*E*G;

if D>0
    Dsq = sqrt(D);
else
    OdhadPolohyBancroft = NaN*ones(4,1);
end

L1 = (-2*F+Dsq)/(2*E);
L2 = (-2*F-Dsq)/(2*E);
Sr1 = Bplus*(a+L1*e);

% Funkce pro výpočet odhadu polohy - Bancroft (vstupem jsou pozice družic a příslušná PSR k nim)
% Počet řádků odpovídá počtu družic v daném čas. okam. (počet sloupců je vždy 3)
% Lorentzova metrika Minkowského časoprostoru (LmMě)
% Matice "B" (pozice družic včetně příslušných PSR)
% Vektor "e" s hodnotami 1
% Vektor "a" s hodnotami NaN

% Cyklus FOR pro výpočet složek vektoru "a"
% Vypočtení složek vektoru "a"
% Konec cyklu

% Součin transponované matice "B" s původní maticí "B"

% Podmínka, zdali je hodnota matice "BTB" rovna 4 (zdali je čtvercová)
% Je-li hodnota "BTB" rovna 4, provede se výpočet matice "B+"

% Pokud není hodnota "BTB" rovna 4, nelze udělat inverzi a výpočet "B+" neproběhne
% Konec podmínky

% Vypočtení vektoru "u"
% Vypočtení vektoru "v"
% Substituce pro výpočet LmMě vektoru "u"
% Substituce pro výpočet LmMě vektorů "u" a "v"
% Substituce pro výpočet LmMě vektoru "v"
% Vypočtení determinantu kvadratické rovnice  $L^2 * E + 2 * L * F + G = 0$ 

% Podmínka, zdali je determinant kladný
% Pokud je determinant kladný, provede se jeho odmocnina

% Pokud není determinant kladný, nemá reálné kořeny (zobrazí se daná věta)
% Konec podmínky

% Vypočtení kořenu "L1" kvadratické rovnice
% Vypočtení kořenu "L2" kvadratické rovnice
% Vypočtení vektoru "Sr1" neznámých paramterů
```

```

Sr2 = Bplus*(a+L2*e); % Vypočtení vektoru "Sr2" neznámých paramterů
res_Sr1 = abs(sum(sqrt((Sr1(1)-PD(:,1)).^2+(Sr1(2)-... % Vypočtení abs. hodnoty sumace reziduí mezi vypočtenou a naměřenou PSR pro "Sr1"
    PD(:,2)).^2+(Sr1(3)-PD(:,3)).^2)+Sr1(4)-PSRsum));
res_Sr2 = abs(sum(sqrt((Sr2(1)-PD(:,1)).^2+(Sr2(2)-... % Vypočtení abs. hodnoty sumace reziduí mezi vypočtenou a naměřenou PSR pro "Sr2"
    PD(:,2)).^2+(Sr2(3)-PD(:,3)).^2)+Sr2(4)-PSRsum));

if res_Sr1 < res_Sr2 % Podmínka, zdali jsou rezidua pro "Sr1" menší než pro "Sr2"
    OdhadPolohyBancroft = Sr1; % Pokud jsou rezidua pro "Sr1" menší, bude "Sr1" vektor neznámých parametrů
else
    OdhadPolohyBancroft = Sr2; % Pokud jsou rezidua pro "Sr1" větší, bude "Sr2" vektor neznámých parametrů
end % Konec podmínky

OdhadPolohyBancroft = OdhadPolohyBancroft'; % Výsledek je transponován do řádkového vektoru
end % Konec funkce

% FUNKCE NEWTON
% -----
% Funkce spočte odhad polohy dle Newtonovy
% iterační metody spolu s KM
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016

```

## FUNKCE\_NEWTON

```

function [OdhadPolohyNewton,KovMatNewton] = Funkce_Newton(PD,PSRsum,STDsumu,S0) % Funkce pro výpočet odhadu polohy (vstupem jsou pozice družic, příslušná PSR, STDsumu a poč. podm.)
[PocetDruzic,n] = size(PD); % Počet řádků odpovídá počtu družic v daném čas. okam. (počet sloupců je vždy 3)
IterMax = 10; % Maximální počet iterací
Iter = 0; % Počáteční hodnota iterace (zatím žádná neproběhla)
Tol = 1e-10; % Tolerance zastavení (pokud hodnota korekcí bude pod tolerancí, nastane konec algoritmu)
Norma = 2*Tol; % Počáteční volba korekcí (na poč. musí být větší než tolerance, aby proběhla iterace)
P0 = NaN*ones(PocetDruzic,1); % Vektor počátečního odhadu pseudovzdáleností "P0" s hodnotami NaN
L = NaN*ones(PocetDruzic,1); % Vektor observací "L" s hodnotami NaN
A = NaN*ones(PocetDruzic,4); % Matice směrových kosinů "A" s hodnotami NaN

while Norma >= Tol && Iter <= IterMax % Cyklus WHILE s dvěma podmínkami, které musí platit současně
    Iter = Iter+1; % Čítač iterací

    for n = 1:PocetDruzic % Vnořený cyklus FOR od 1 do počtu družic v daném čas. okam. ("n" značí index družice)
        P0(n,1) = sqrt((S0(1)-PD(n,1))^2+(S0(2)-PD(n,2))^2+(S0(3)-PD(n,3))^2)+S0(4); % Vypočtení vektoru počátečního odhadu pseudovzdáleností "P0"
        L(n,1) = PSRsum(n)-P0(n); % Vypočtení vektoru observací "L"
        A(n,1) = (S0(1)-PD(n,1))/sqrt((S0(1)-PD(n,1))^2+(S0(2)-PD(n,2))^2+(S0(3)-PD(n,3))^2); % Vypočtení matice směrových kosinů "A"
        A(n,2) = (S0(2)-PD(n,2))/sqrt((S0(1)-PD(n,1))^2+(S0(2)-PD(n,2))^2+(S0(3)-PD(n,3))^2);
        A(n,3) = (S0(3)-PD(n,3))/sqrt((S0(1)-PD(n,1))^2+(S0(2)-PD(n,2))^2+(S0(3)-PD(n,3))^2);
        A(n,4) = 1;
    end
end

```

```

end

ATA = A'*A;

if rank(ATA) == 4
    DeltaS = inv(ATA)*(A'*L);
    S0 = S0+DeltaS';
    Norma = norm(DeltaS);
else
    S0 = Inf;
end

end

% Konec cyklu FOR

% Součin transponované matice "A" s původní maticí "A"

% Podmínka, zdali je hodnota matice "ATA" rovna 4 (zdali je čtvercová)
% Je-li hodnota ATA rovna 4, proběhne výpočet "DeltaS"
% Korekce souřadnic a času o "DeltaS" (to je transponované na řádkový vektor)
% Výpočet normy vektoru "DeltaS"

% Pokud není hodnota "ATA" rovna 4, výpočet "DeltaS" neproběhne
% Konec podmínky

end

% Konec cyklu WHILE

% Výsledný odhad souřadnic a časové diference
% Výpočet kovarianční matice

end

% Konec funkce

% FUNKCE KALMAN R2
% -----
% Funkce spočte odhad polohy dle Kalmanova
% rozšířeného filtru pro 2.řád modelu včetně KM
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016

```

## FUNKCE\_KALMAN PRO 2.ŘÁD MODELU

```

function [X,P] = Funkce_KalmanR2(PD,PSRsum,STDsumu,T,Xi,Pi)
I = eye(8);
Cv = 5e04;
Ca = 5;
QS = [T^3/3,T^2/2;T^2/2,T];
Qsb = [0,0;0,0];
Q = blkdiag(Ca*QS,Ca*QS,Ca*QS,Cv*Qsb);
[PocetDruzic,~] = size(PD);
R = STDsumu*STDsumu*eye(PocetDruzic);
gXp = NaN*ones(PocetDruzic,1);
H = NaN*ones(PocetDruzic,8);
Z = PSRsum;
FS = [1,T;0,1];
FSb = [0,0;0,0];
F = blkdiag(FS,FS,FS,FSb);

% Funkce pro výpočet odhadu polohy (vstupem jsou pozice družic, příslušná PSR, STDsumu, T, X a P)
% Jednotková matice
% Spektrální výkonová hustota rychlosti
% Spektrální výkonová hustota zrychlení

% Kovarianční matice šumu stavu (Process Noise CovMat)
% Počet řádků odpovídá počtu družic v daném čas. okam. (počet sloupců je vždy 3)
% Kovarianční matice šumu měření (Measurement Noise CovMat)
% Vektor pseudovzdáleností s hodnotami NaN
% Matice měření s hodnotami NaN (Measurement Matrix)
% Vektor měření

% Matice dynamiky / přechodu (State Transition Matrix)

```

```

Xp = F*Xi; % Predikce stavového vektoru

for n = 1:PocetDruzic % Cyklus FOR od 1 do počtu družic v daném čas. okam. ("n" značí index družice)
    gXp(n,1) = sqrt((Xp(1)-PD(n,1))^2+(Xp(3)-PD(n,2))^2+(Xp(5)-PD(n,3))^2)+Xp(7); % Vypočtení pseudovzdálenosti
    H(n,1) = (Xp(1)-PD(n,1))/sqrt((Xp(1)-PD(n,1))^2+(Xp(3)-PD(n,2))^2+(Xp(5)-PD(n,3))^2); % Vypočtení matice měření
    H(n,3) = (Xp(3)-PD(n,2))/sqrt((Xp(1)-PD(n,1))^2+(Xp(3)-PD(n,2))^2+(Xp(5)-PD(n,3))^2);
    H(n,5) = (Xp(5)-PD(n,3))/sqrt((Xp(1)-PD(n,1))^2+(Xp(3)-PD(n,2))^2+(Xp(5)-PD(n,3))^2);
    H(n,7) = 1;
    H(n,[2,4,6,8]) = 0;
end % Konec cyklu FOR

Pp = F*Pi*F'+Q; % Predikce kovarianční matice chyb stavu
K = Pp*H'/(H*Pp*H'+R); % Vypočtení Kalmanova zisku (gain)
X = Xp+K*(Z-gXp); % Filtrace stavového vektoru
P = (I-K*H)*Pp; % Filtrace kovarianční matice chyb stavu
end % Konec funkce

% FUNKCE KALMAN R3
% -----
% Funkce spočte odhad polohy dle Kalmanova
% rozšířeného filtru pro 3.řád modelu včetně KM
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016

```

### FUNKCE\_KALMAN PRO 3.ŘÁD MODELU

```

function [X,P] = Funkce_KalmanR3(PD,PSRsum,STDsumu,T,Xi,Pi) % Funkce pro výpočet odhadu polohy (vstupem jsou pozice družic, příslušná PSR, STDsumu, T, X a P)
I = eye(12); % Jednotková matice
Ca = 5; % Spektrální výkonová hustota zrychlení
Cd = 5e-04; % Spektrální výkonová hustota driftu
QS = [T^5/20,T^4/8,T^3/6;T^4/8,T^3/3,T^2/2;T^3/6,T^2/2,T];
Qsb = [0,0,0;0,0,0;0,0,0];
Q = blkdiag(Cd*QS,Cd*QS,Cd*QS,Ca*Qsb); % Kovarianční matice šumu stavu (Process Noise CovMat)
[PocetDruzic,~] = size(PD); % Počet řádků odpovídá počtu družic v daném čas. okam. (počet sloupců je vždy 3)
R = STDsumu*STDsumu*eye(PocetDruzic); % Kovarianční matice šumu měření (Measurement Noise CovMat)
gXp = NaN*ones(PocetDruzic,1); % Vektor pseudovzdáleností s hodnotami NaN
H = NaN*ones(PocetDruzic,12); % Matice měření s hodnotami NaN (Measurement Matrix)
Z = PSRsum; % Vektor měření
FS = [1,T,T^2/2;0,1,T;0,0,1];
FSb = [0,0,0;0,0,0;0,0,0];
F = blkdiag(FS,FS,FS,FSb);
Xp = F*Xi; % Matice dynamiky / přechodu (State Transition Matrix)
% Predikce stavového vektoru

```

```

for n = 1:PocetDruzic % Cyklus FOR od 1 do počtu družic v daném čas. okam. ("n" značí index družice)
    gXp(n,1) = sqrt((Xp(1)-PD(n,1))^2+(Xp(4)-PD(n,2))^2+(Xp(7)-PD(n,3))^2)+Xp(10); % Vypočtení pseudovzdáleností
    H(n,1) = (Xp(1)-PD(n,1))/sqrt((Xp(1)-PD(n,1))^2+(Xp(4)-PD(n,2))^2+(Xp(7)-PD(n,3))^2); % Vypočtení matice měření
    H(n,4) = (Xp(4)-PD(n,2))/sqrt((Xp(1)-PD(n,1))^2+(Xp(4)-PD(n,2))^2+(Xp(7)-PD(n,3))^2);
    H(n,7) = (Xp(7)-PD(n,3))/sqrt((Xp(1)-PD(n,1))^2+(Xp(4)-PD(n,2))^2+(Xp(7)-PD(n,3))^2);
    H(n,10) = 1;
    H(n,[2,3,5,6,8,9,11,12]) = 0;
end % Konec cyklu FOR

Pp = F*Pi*F'+Q; % Predikce kovarianční matice chyb stavu
K = Pp*H'/(H*Pp*H'+R); % Vypočtení Kalmanova zisku (gain)
X = Xp+K*(Z-gXp); % Filtrace stavového vektoru
P = (I-K*H)*Pp; % Filtrace kovarianční matice chyb stavu
end % Konec funkce

% FUNKCE ELIPSOID
% -----
% Funkce ověří správnost odhadu
% -----
% Copyright (c) Bc. Lukáš Diblík, 2016

```

## FUNKCE\_ELIPSOID

```

function [Evaluate] = Funkce_Elipsoid(Stred,KovMat,PST,OverBod) % Funkce pro určení, zda je odhad správný (vstupem je odhad polohy, KM, pravděpodobnost a poloha přijímače)
Vek = Stred-OverBod; % Vypočtení rozdílu hodnot odhadu polohy a skutečné polohy přijímače (středem elipsoidu je odhad polohy)
LevaStrana = Vek*inv(KovMat)*Vek'; % Vypočtení levé strany (KovMat definuje elipsoid (jeho natočení a roztažení))

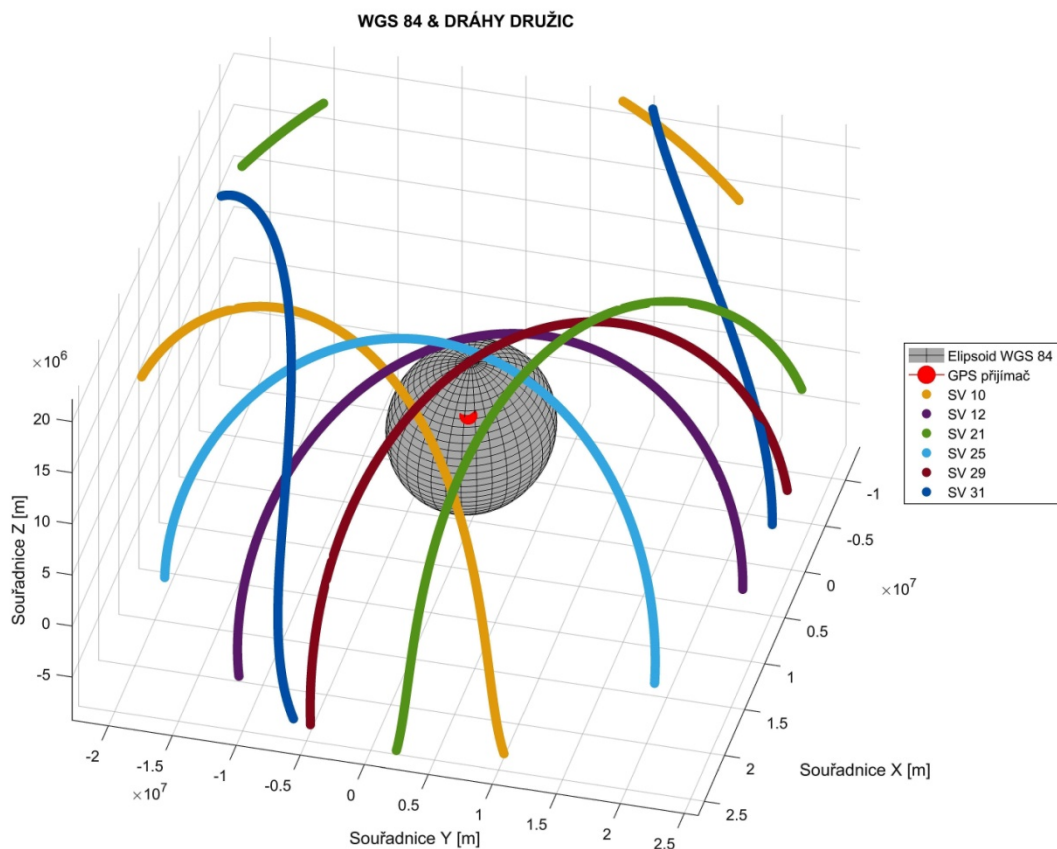
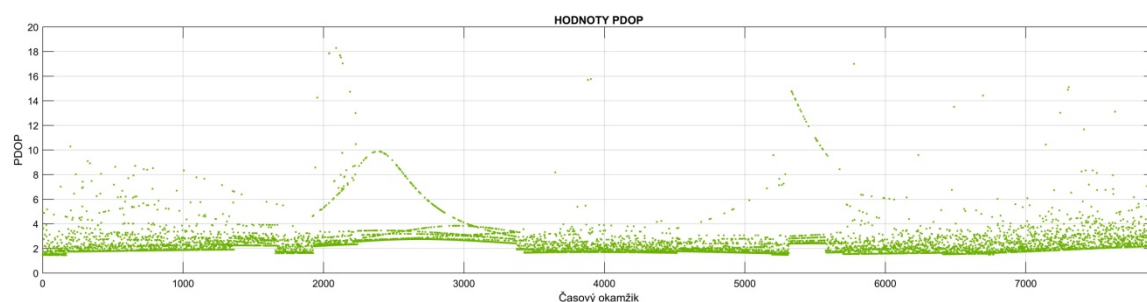
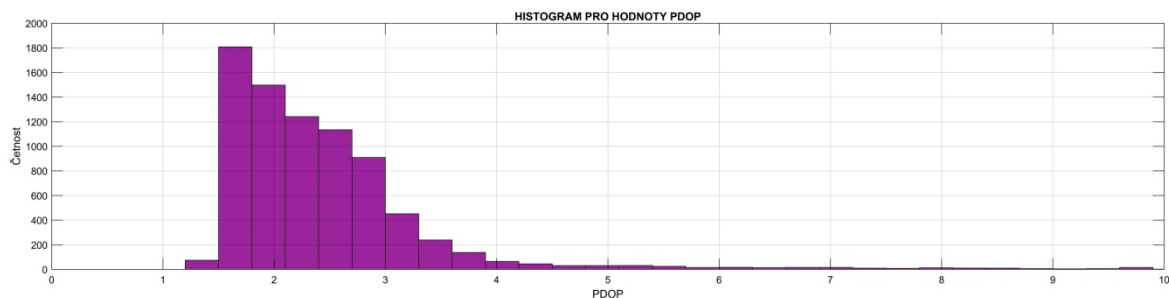
if LevaStrana <= chi2inv(PST,length(Vek)) % Podmínka, zdali je levá strana rovnice menší nebo rovna hranicím elipsoidu (hranice je rovna kvantilu chi2 rozdělení)
    Evaluate = 1; % Pokud poloha přijímače leží uvnitř nebo na hraně elipsoidu, je "Evaluate" rovna 1
else % Pokud poloha přijímače leží vně elipsoidu, je "Evaluate" rovna 0
    Evaluate = 0;
end % Konec podmínky

end % Konec funkce

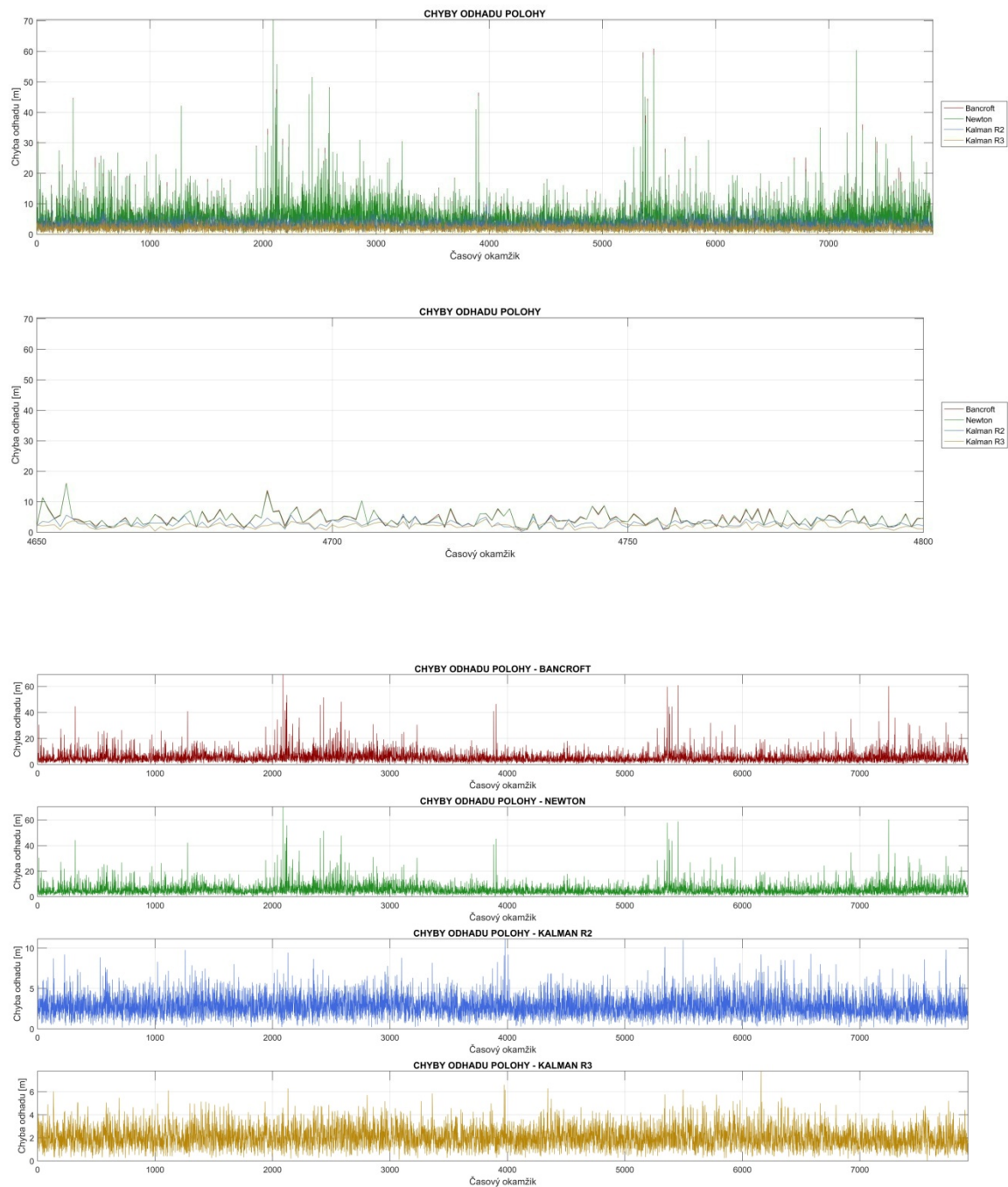
```

# Příloha B: Výstupy provedených simulací

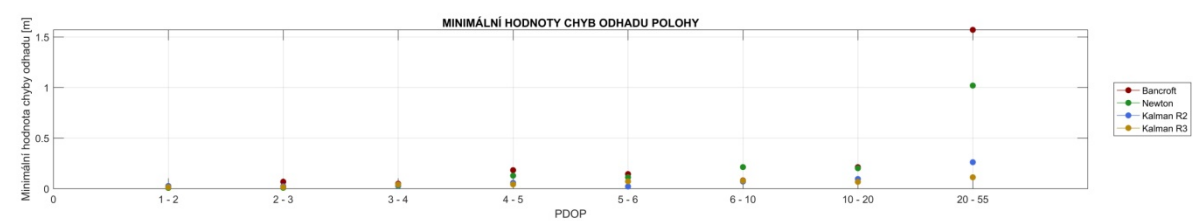
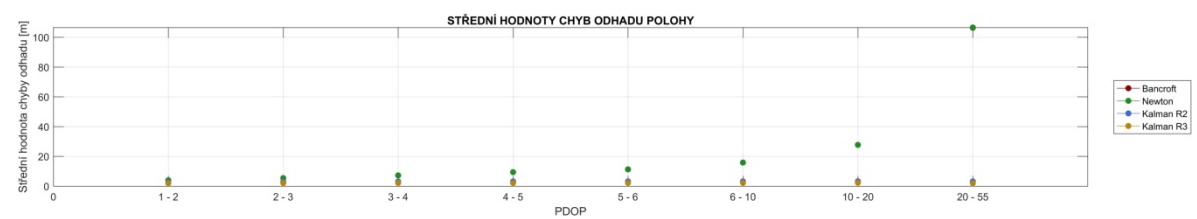
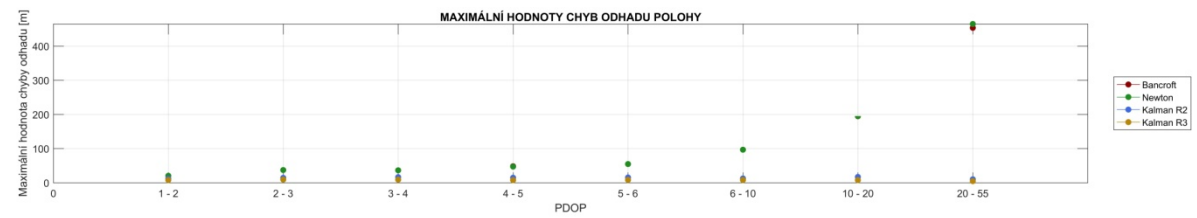
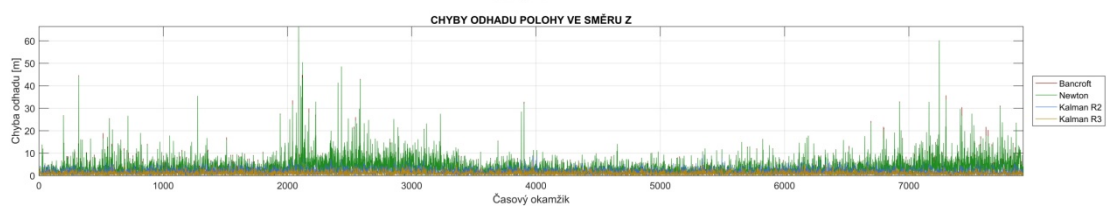
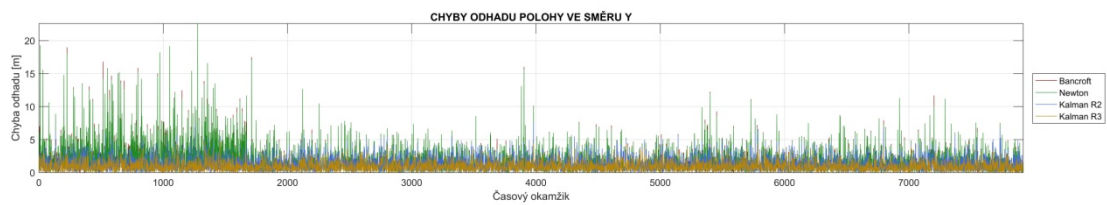
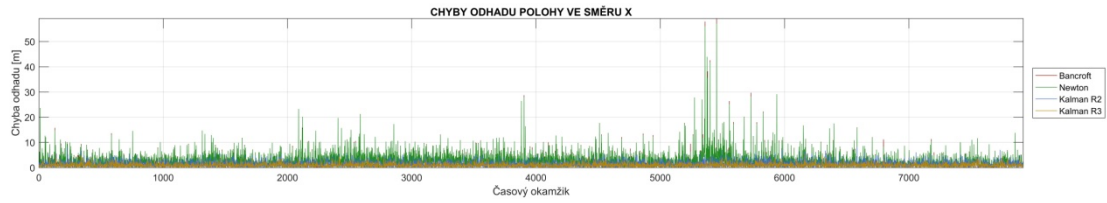
## Grafy společné pro všechny simulace

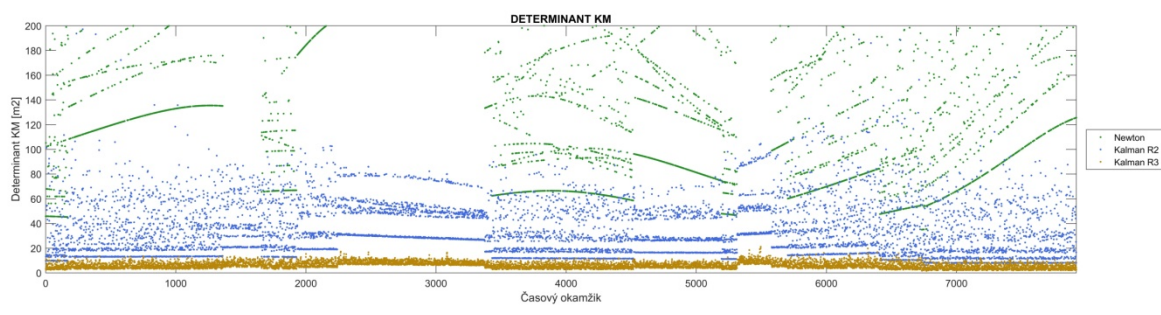
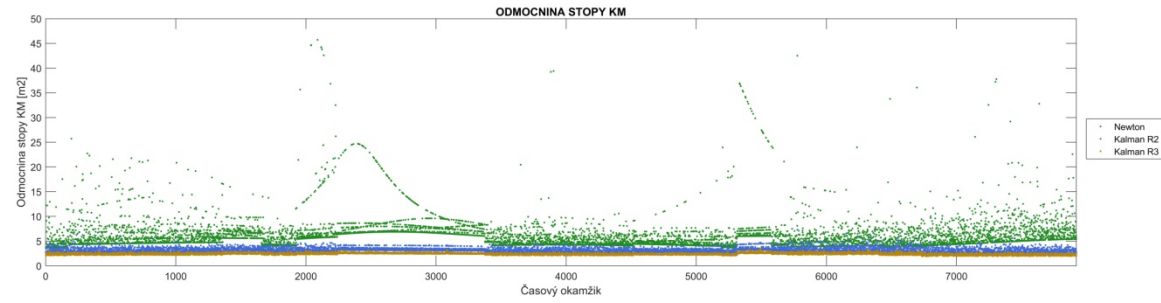
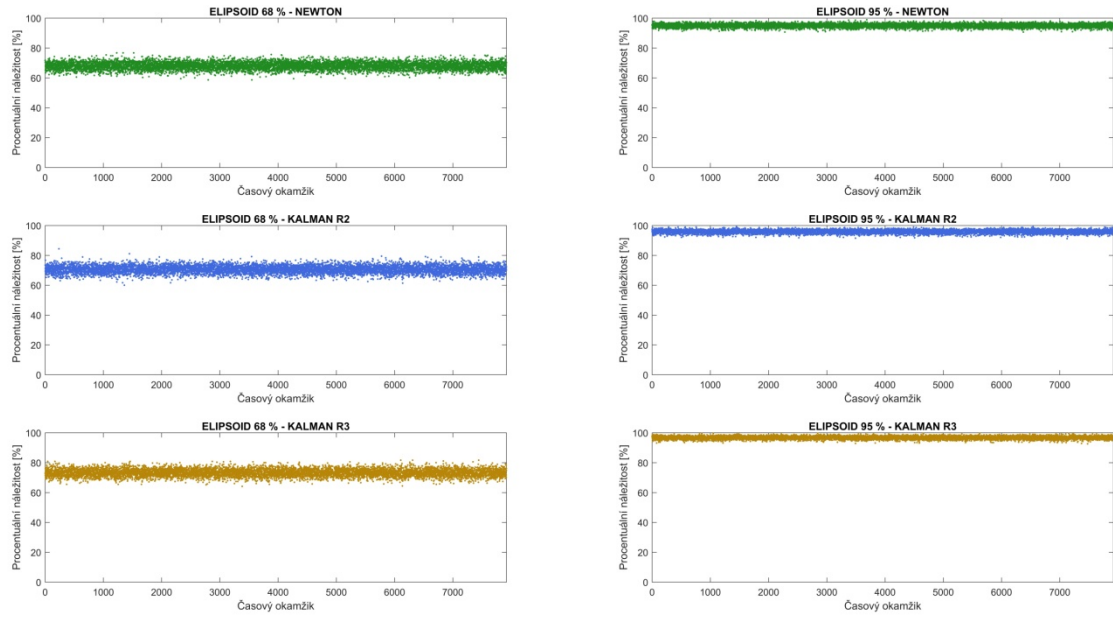


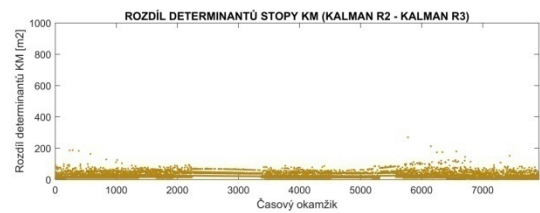
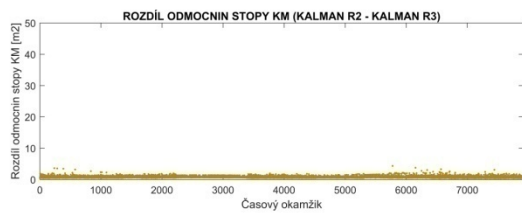
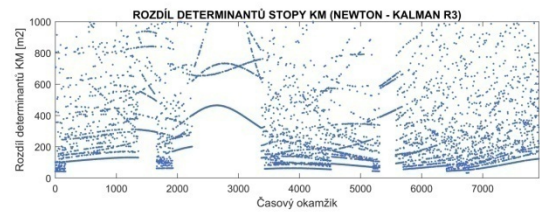
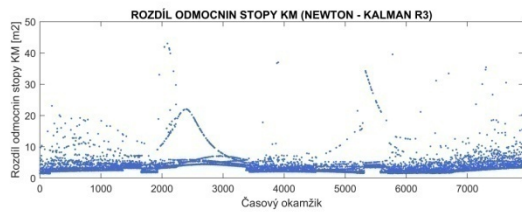
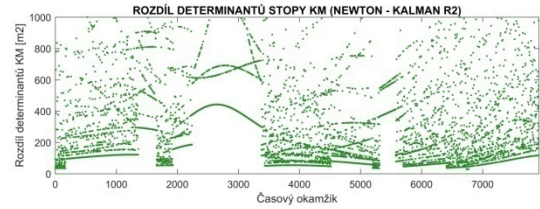
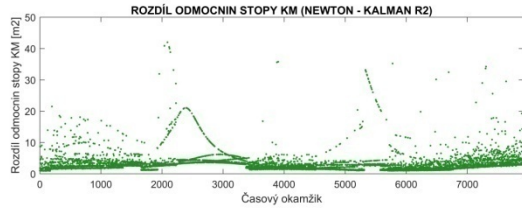
## Simulace 1



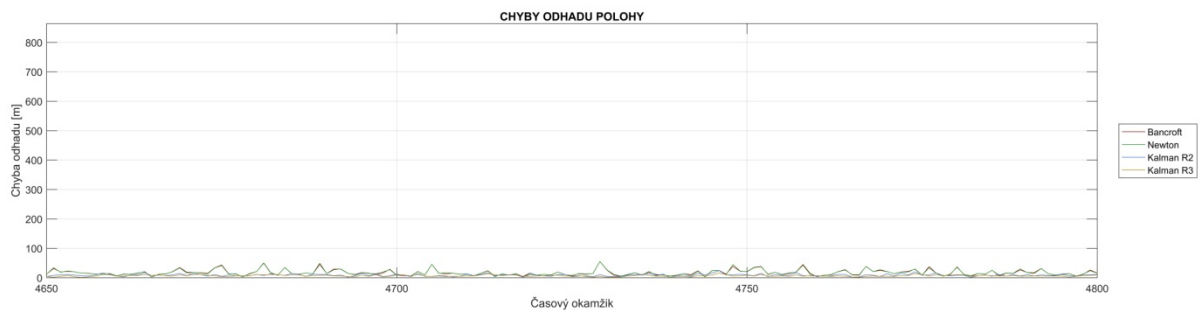
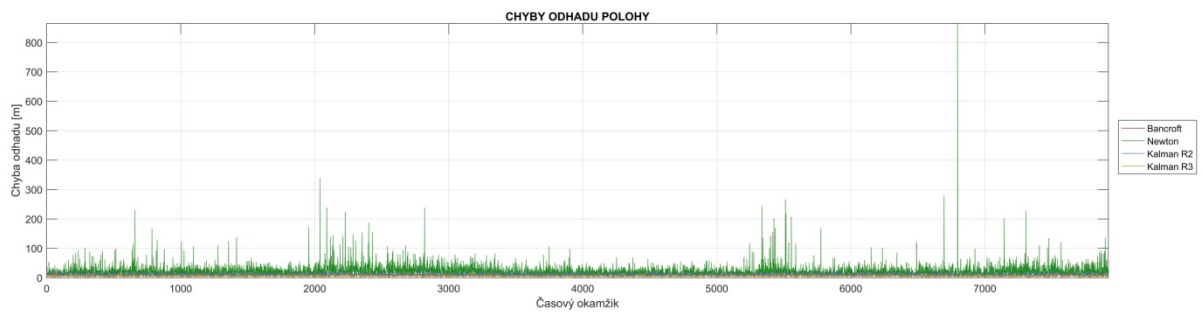


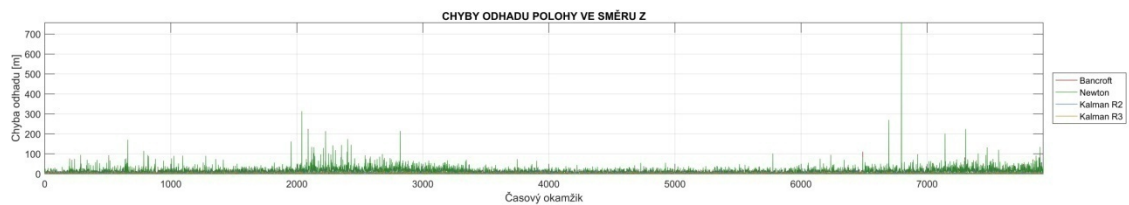
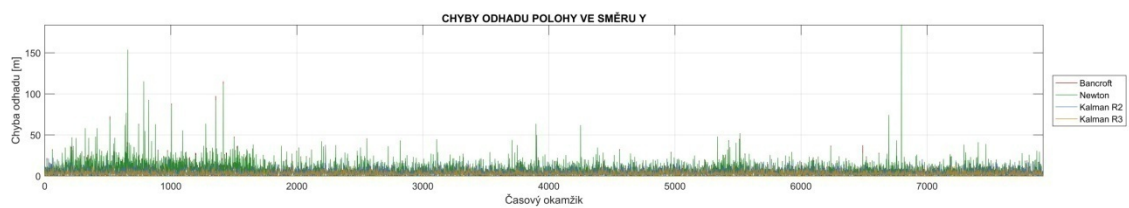
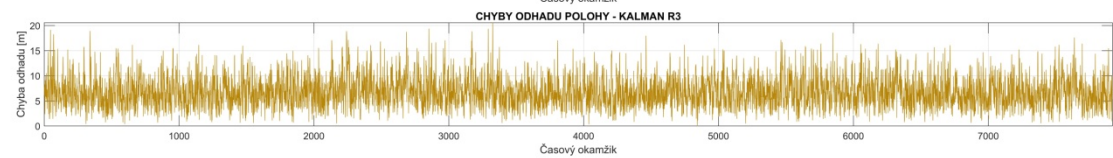
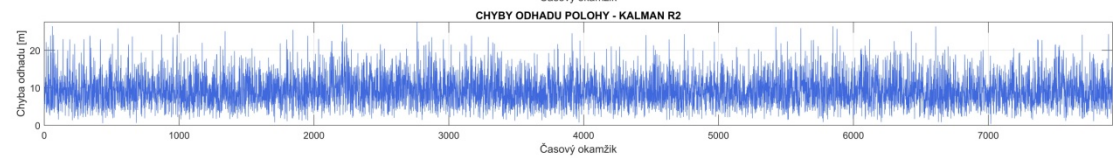
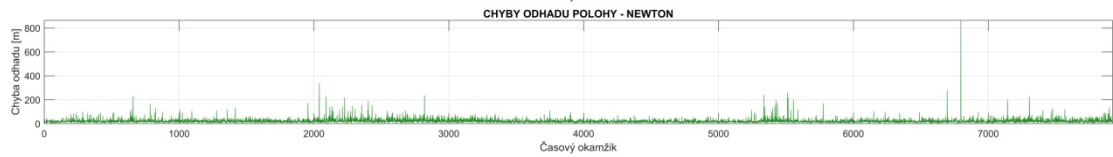


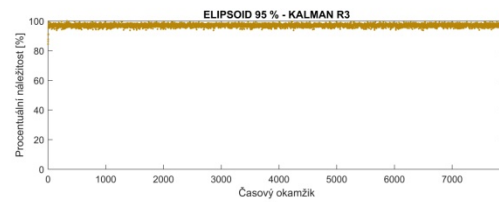
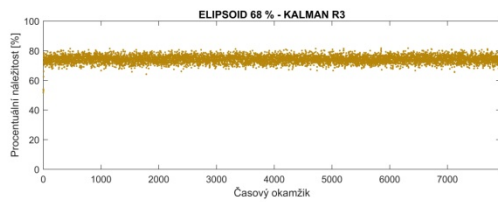
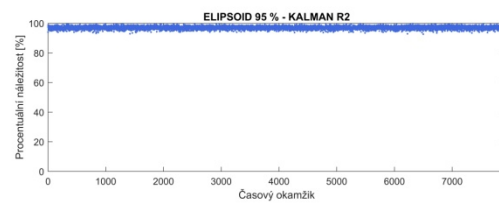
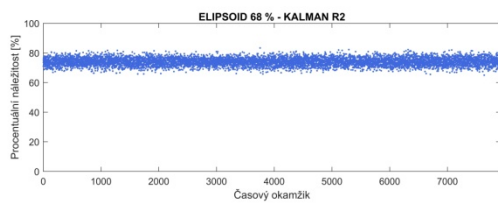
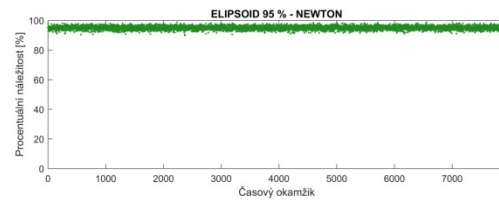
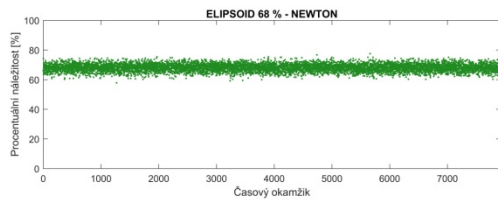
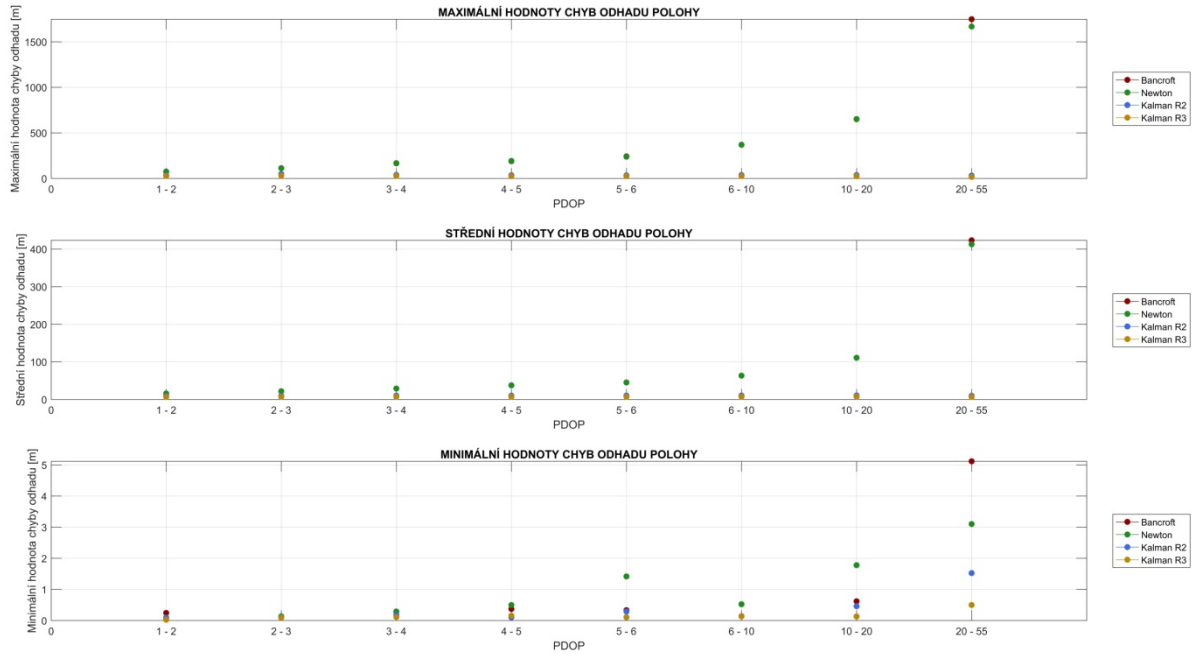


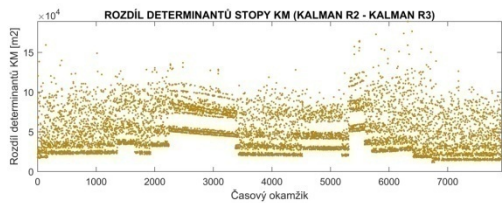
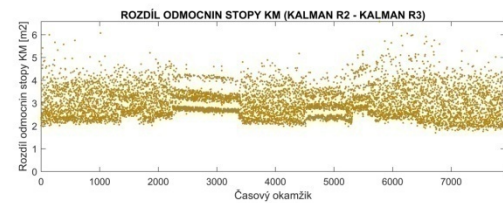
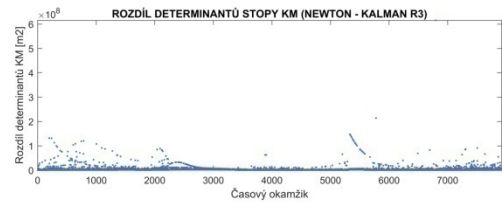
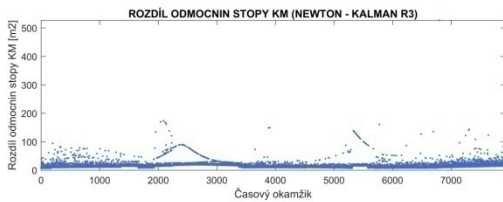
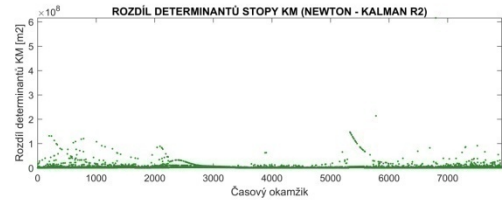
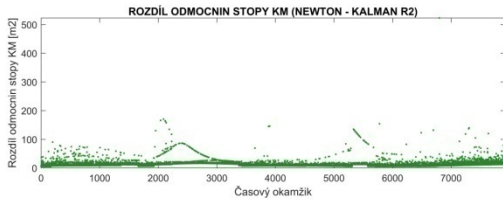
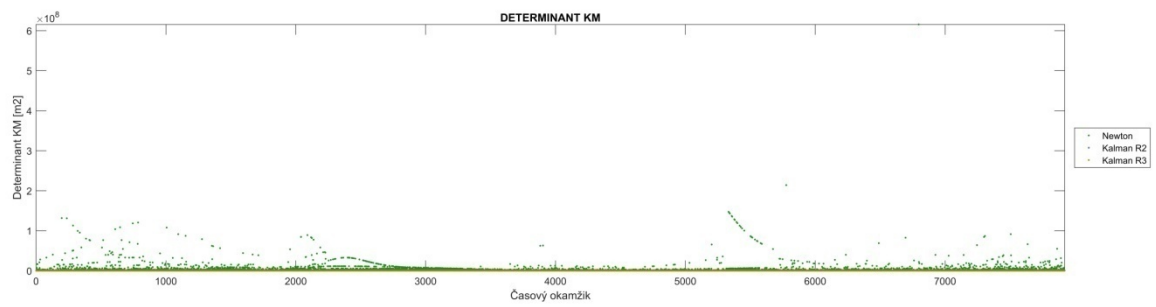
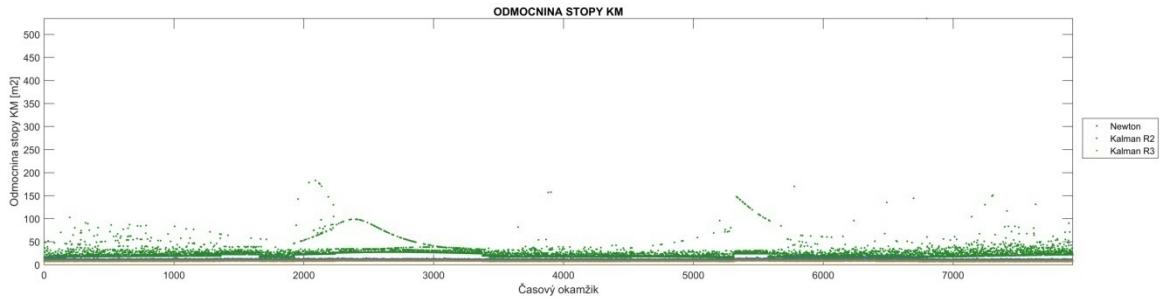


## Simulace 2



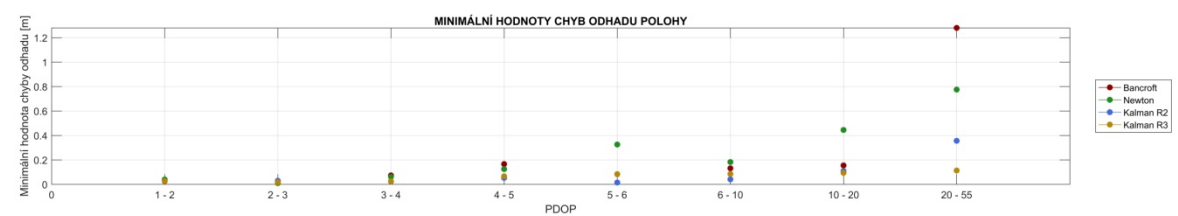
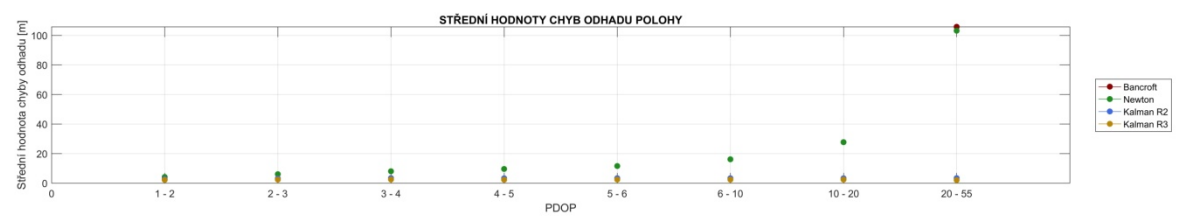
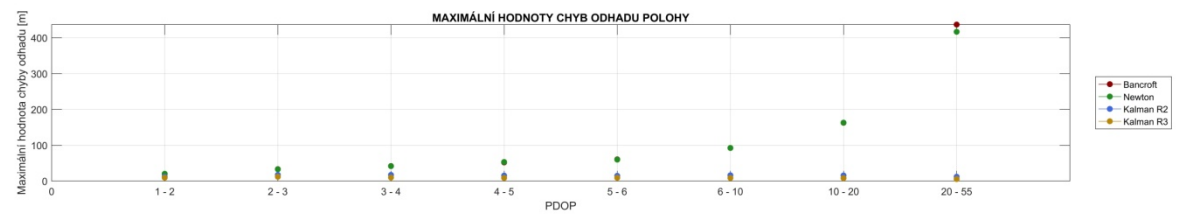
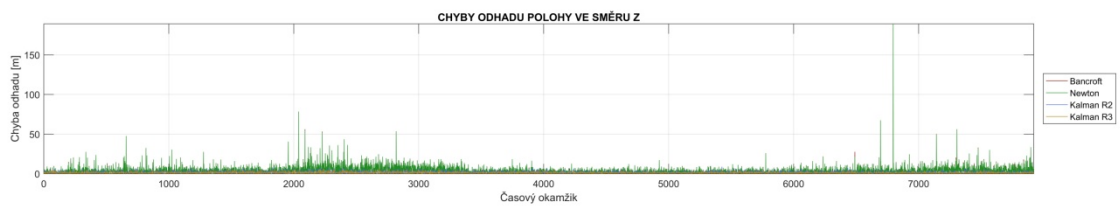
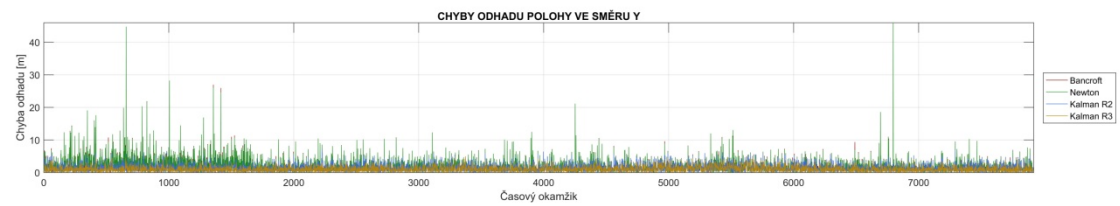
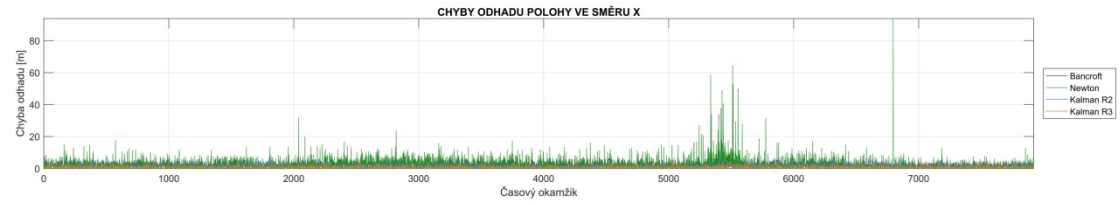




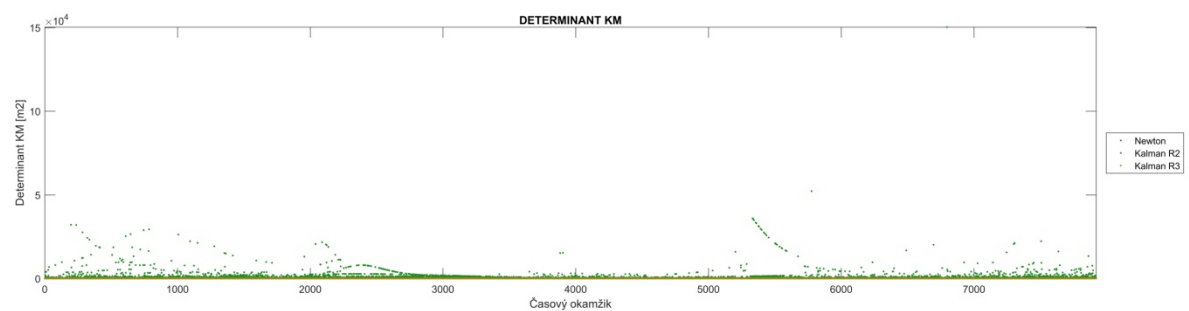
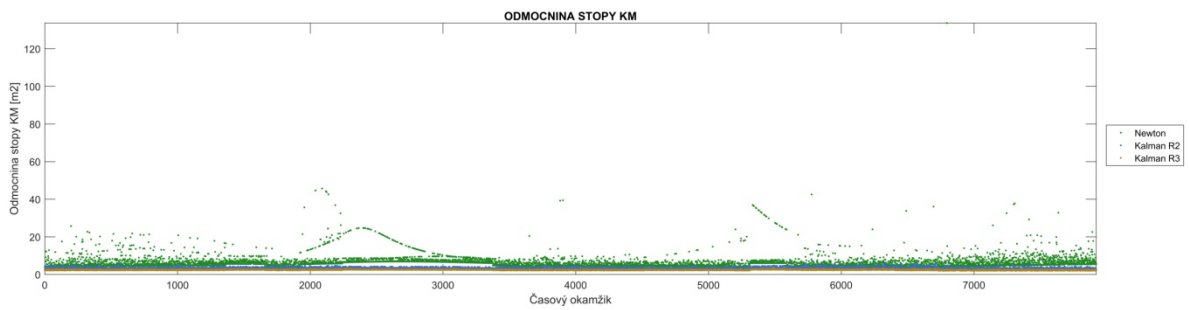
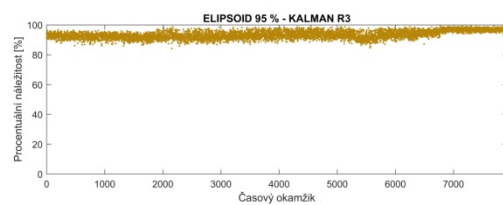
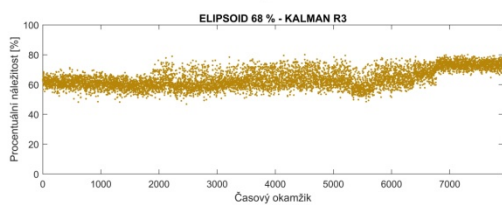
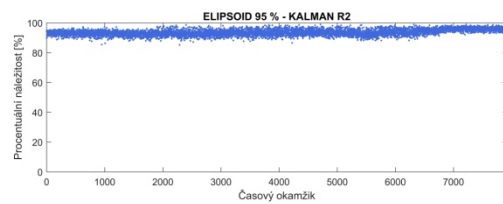
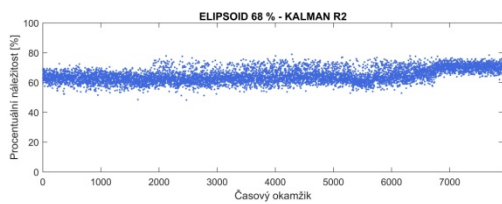
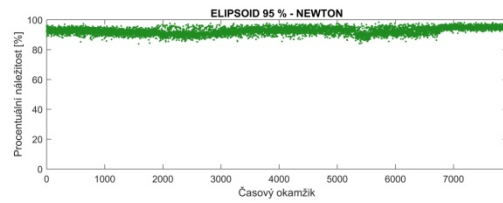
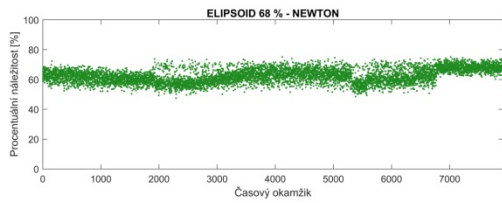


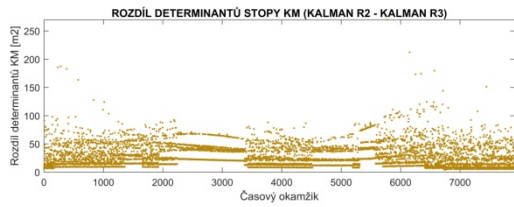
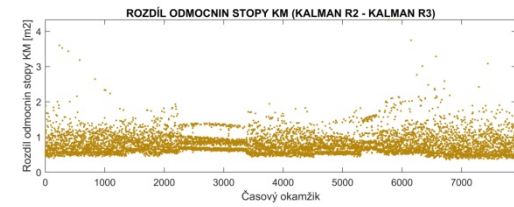
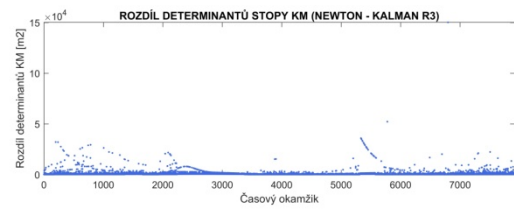
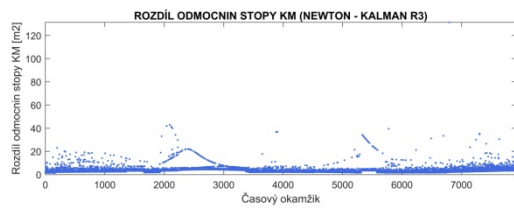
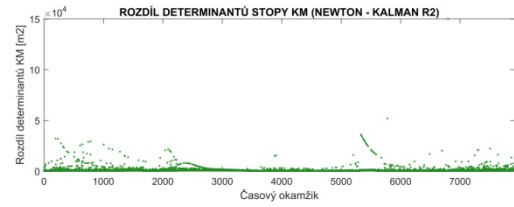
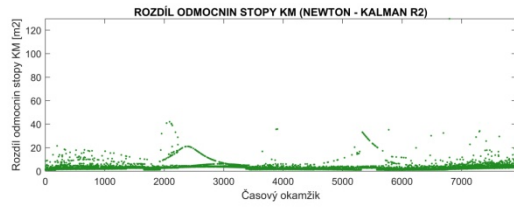
### Simulace 3



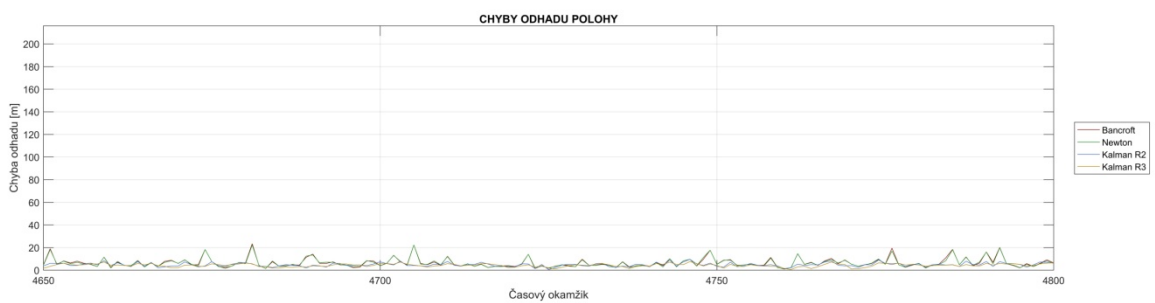
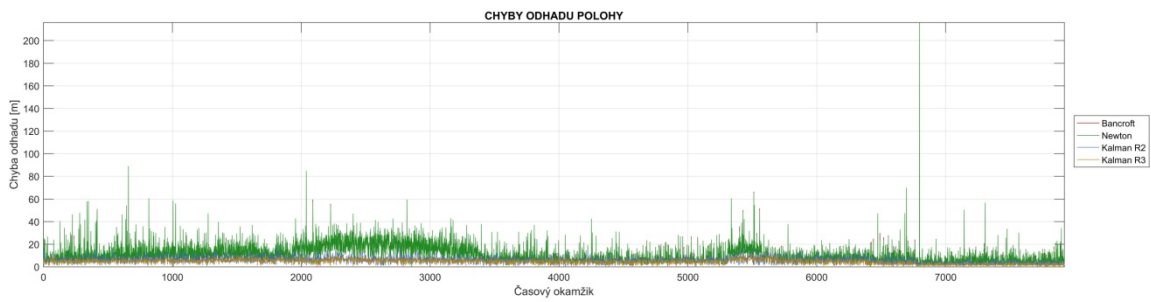


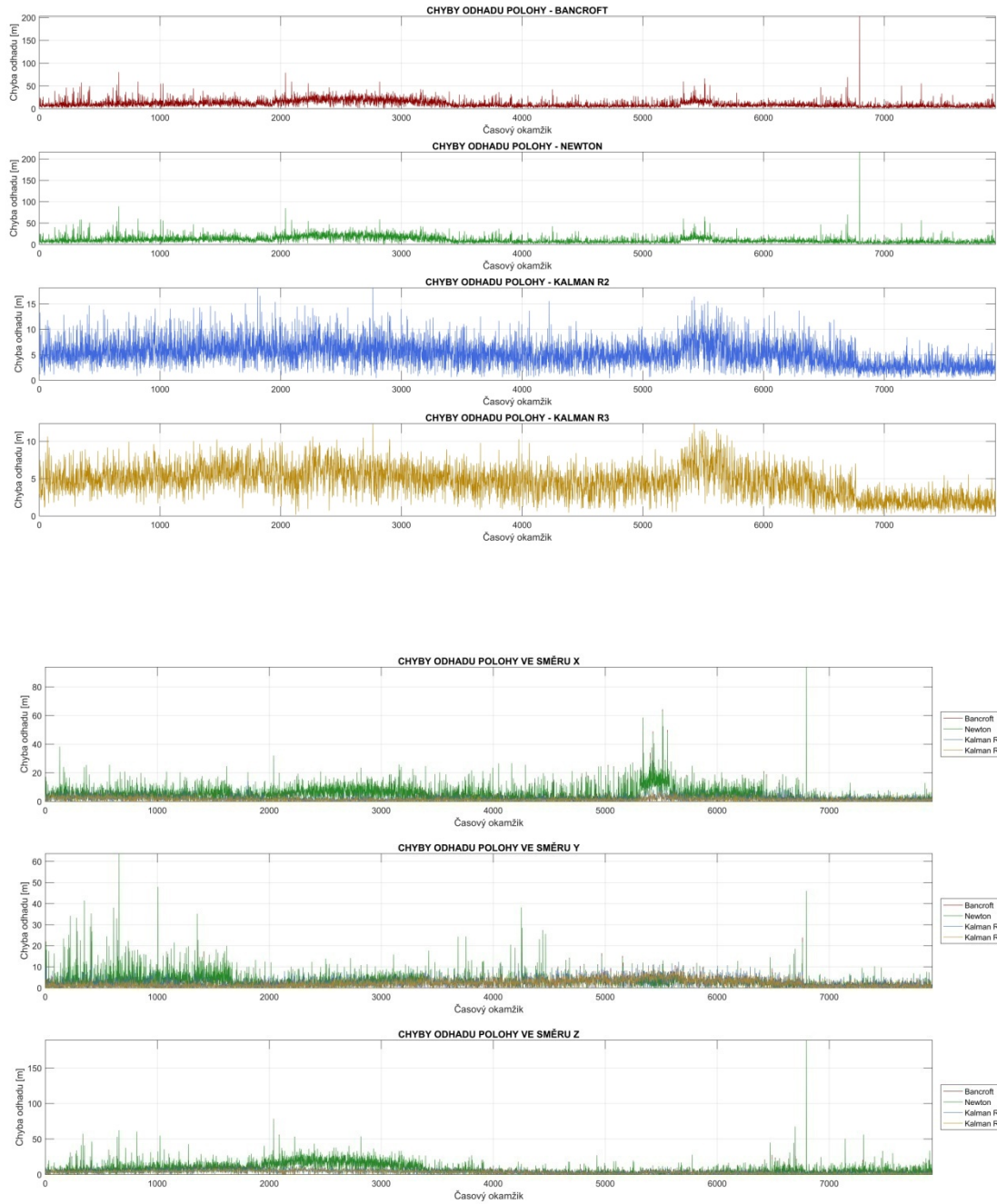


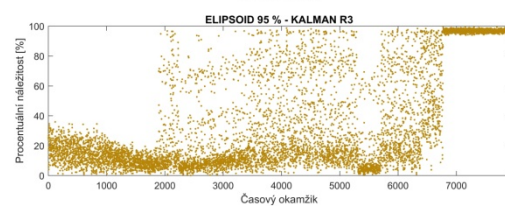
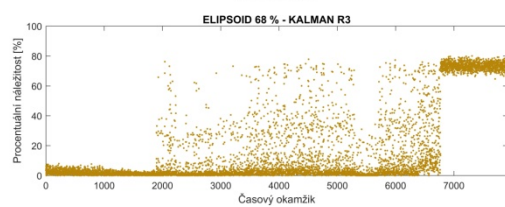
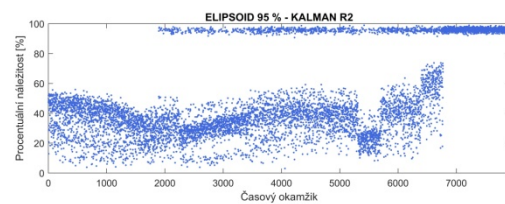
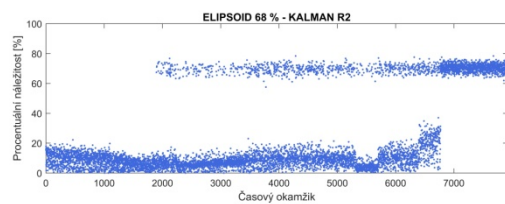
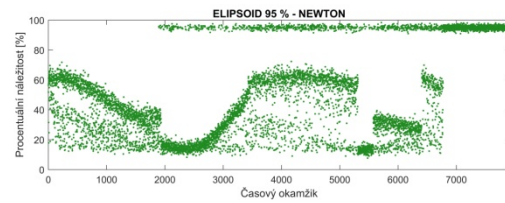
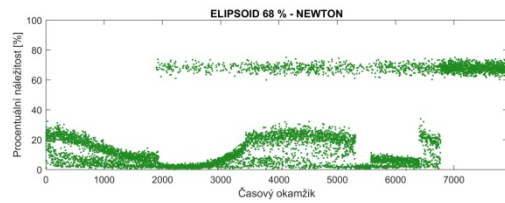
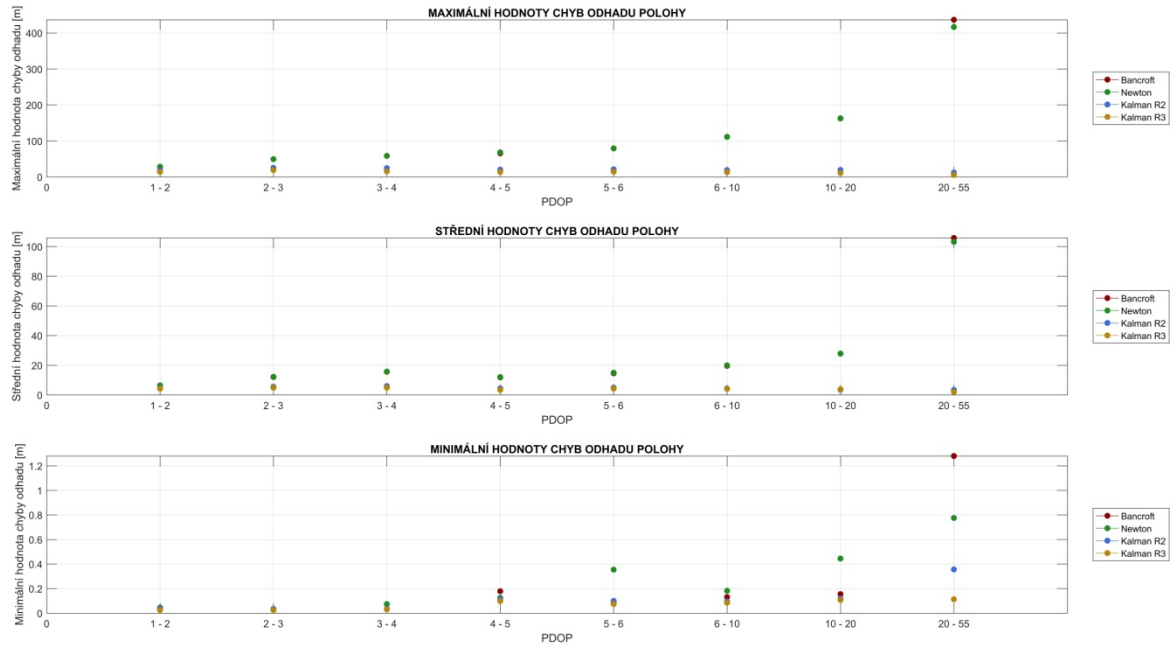


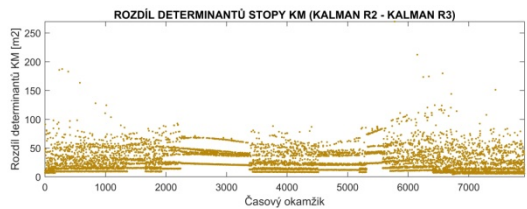
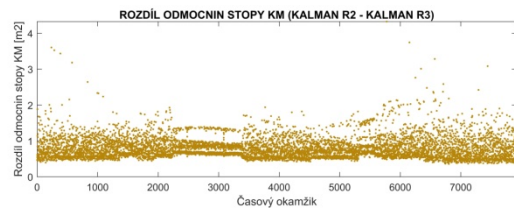
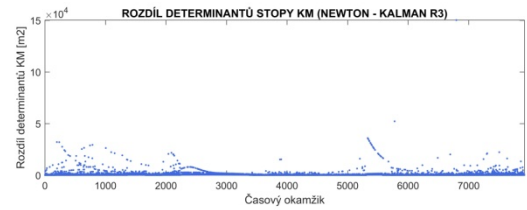
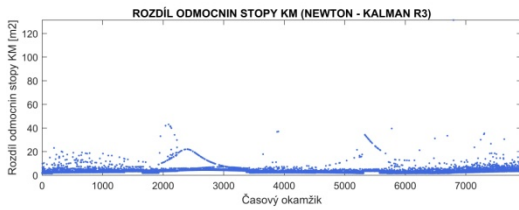
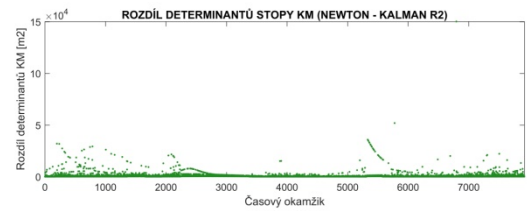
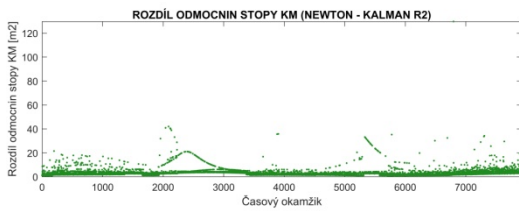
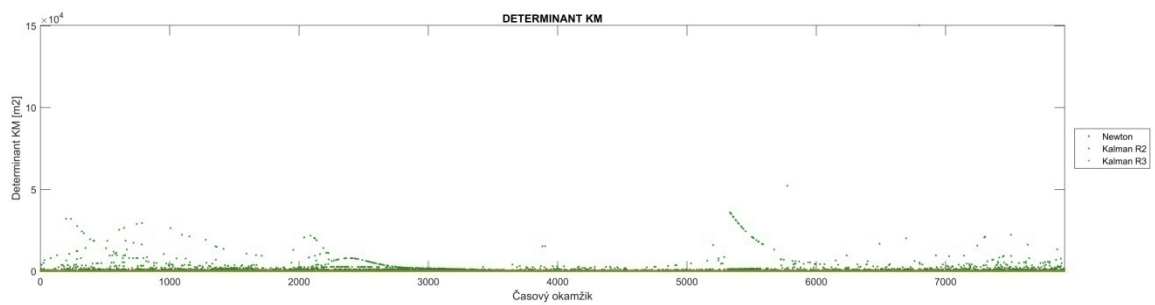
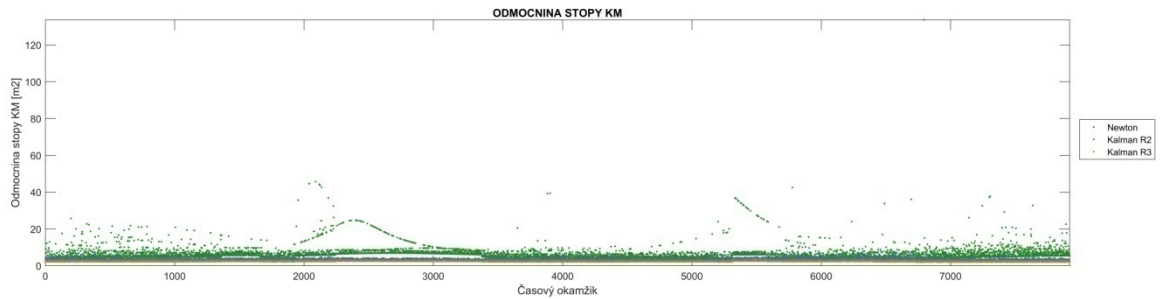


## Simulace 4

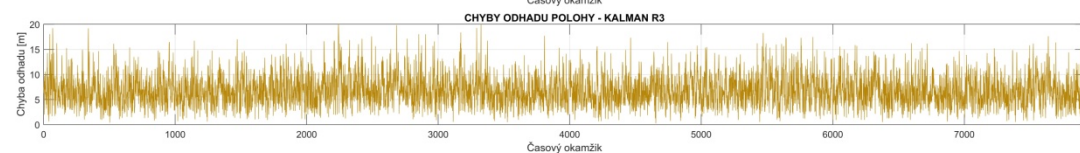
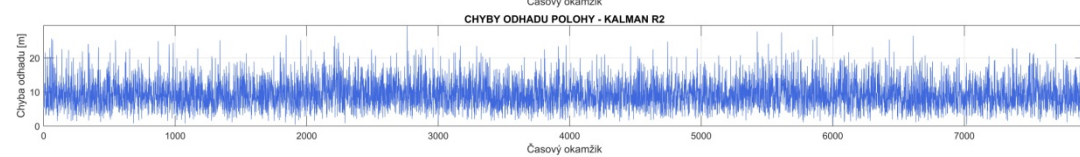
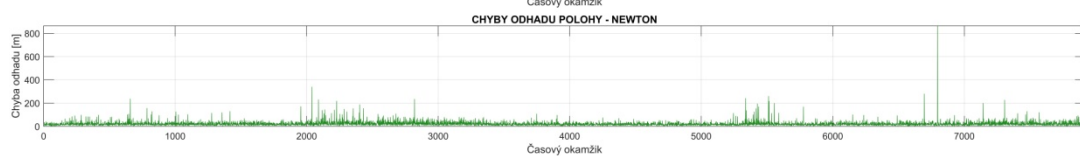
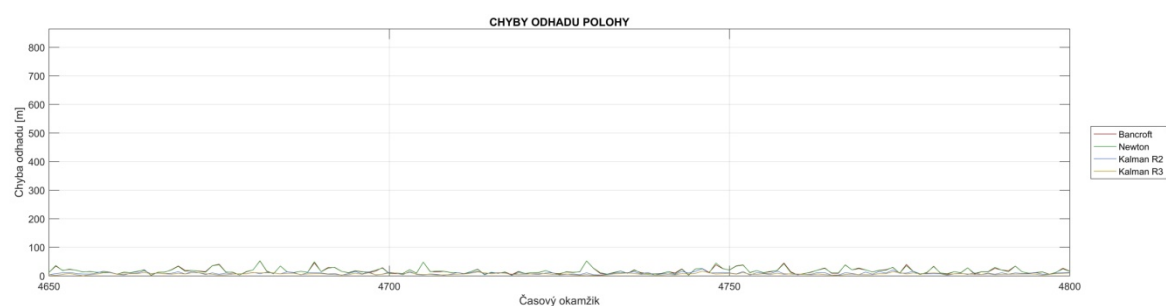
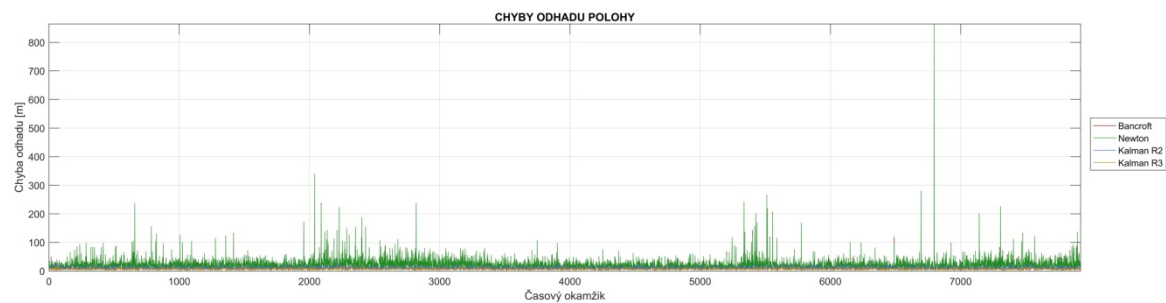


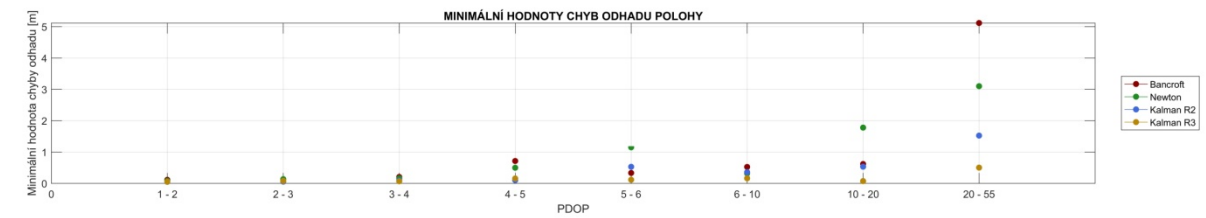
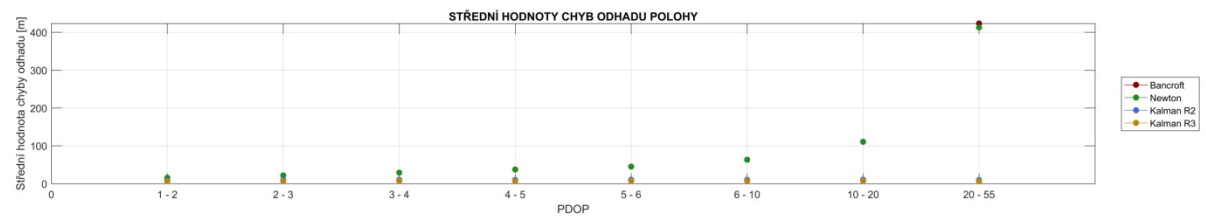
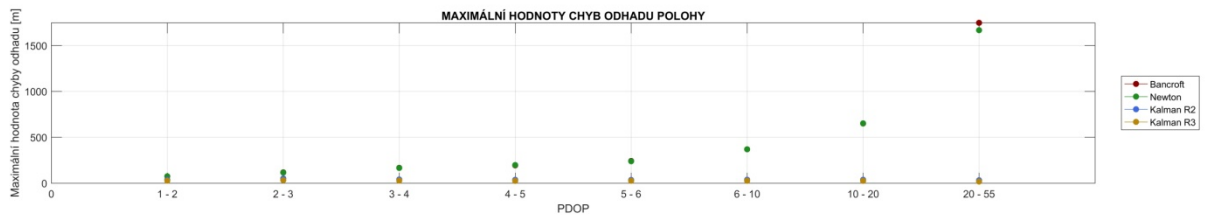
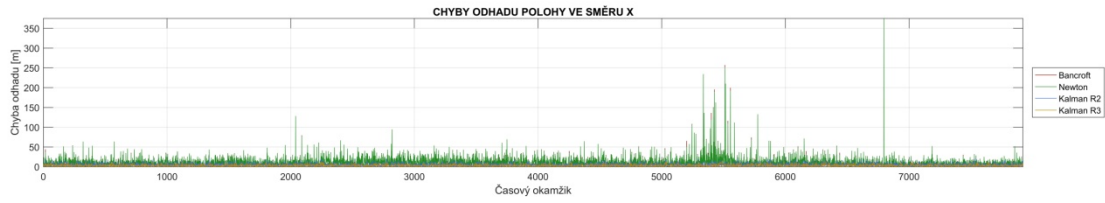


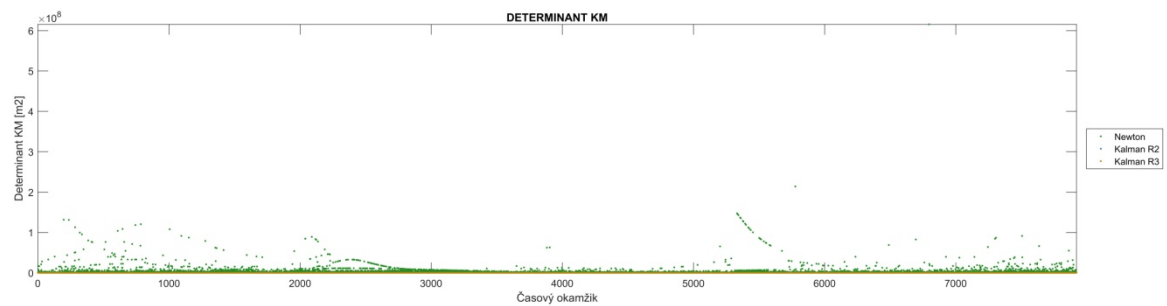
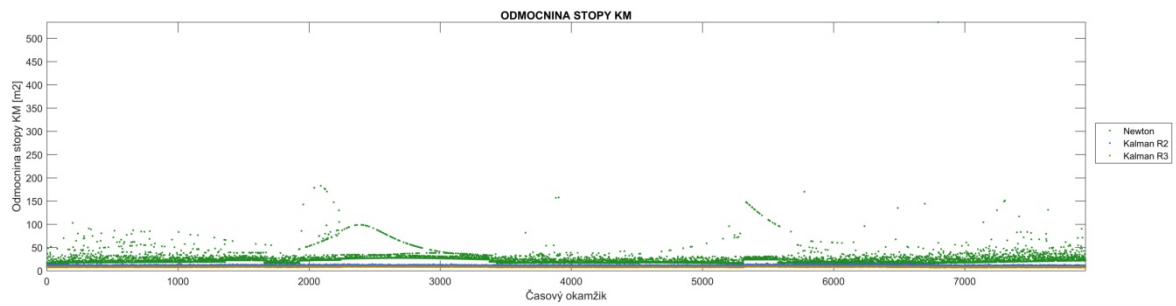
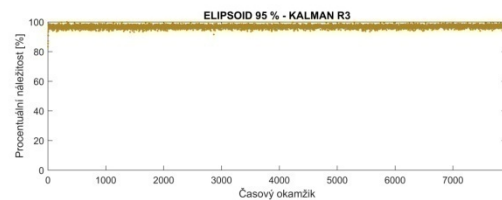
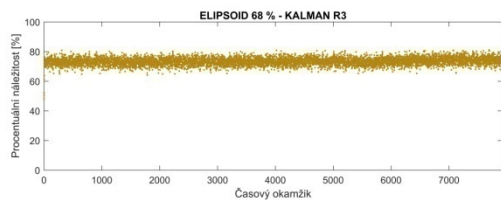
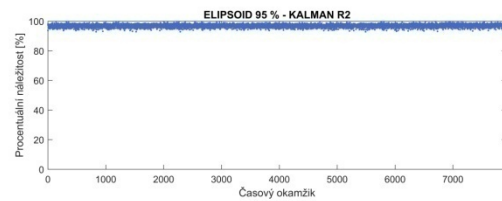
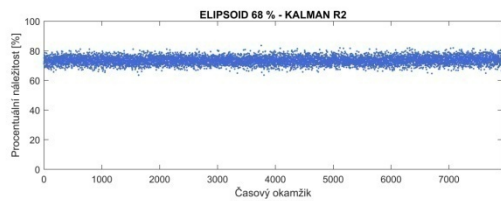
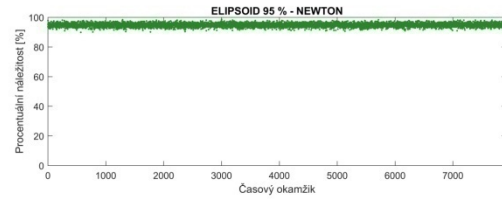
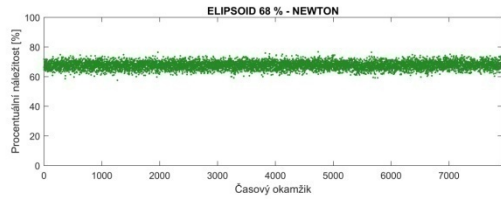




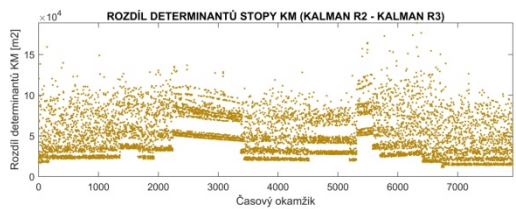
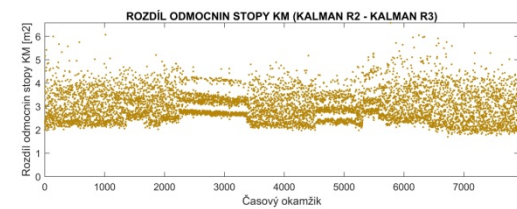
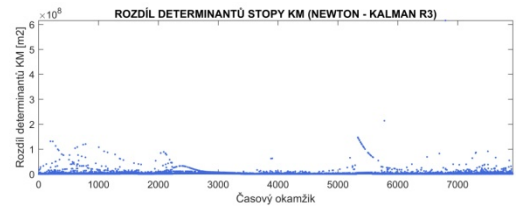
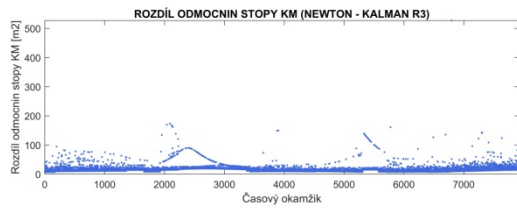
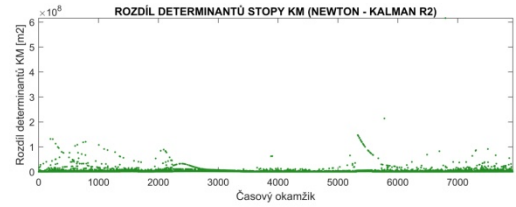
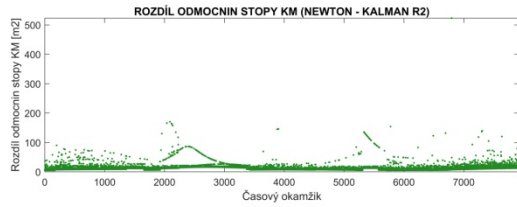
## Simulace 5



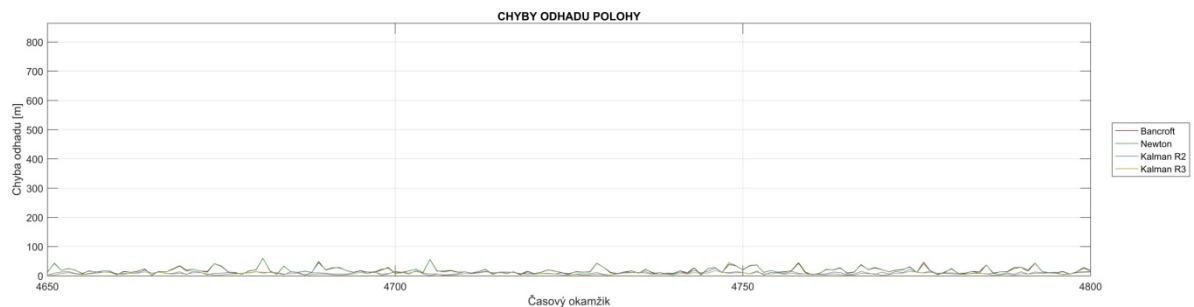
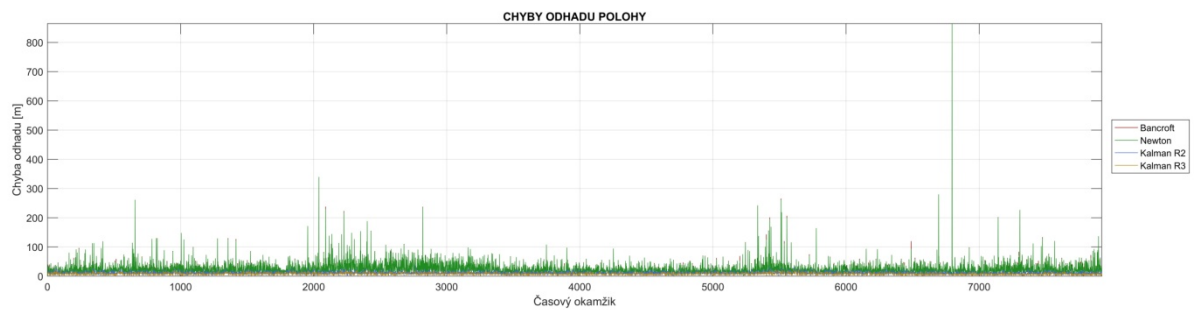


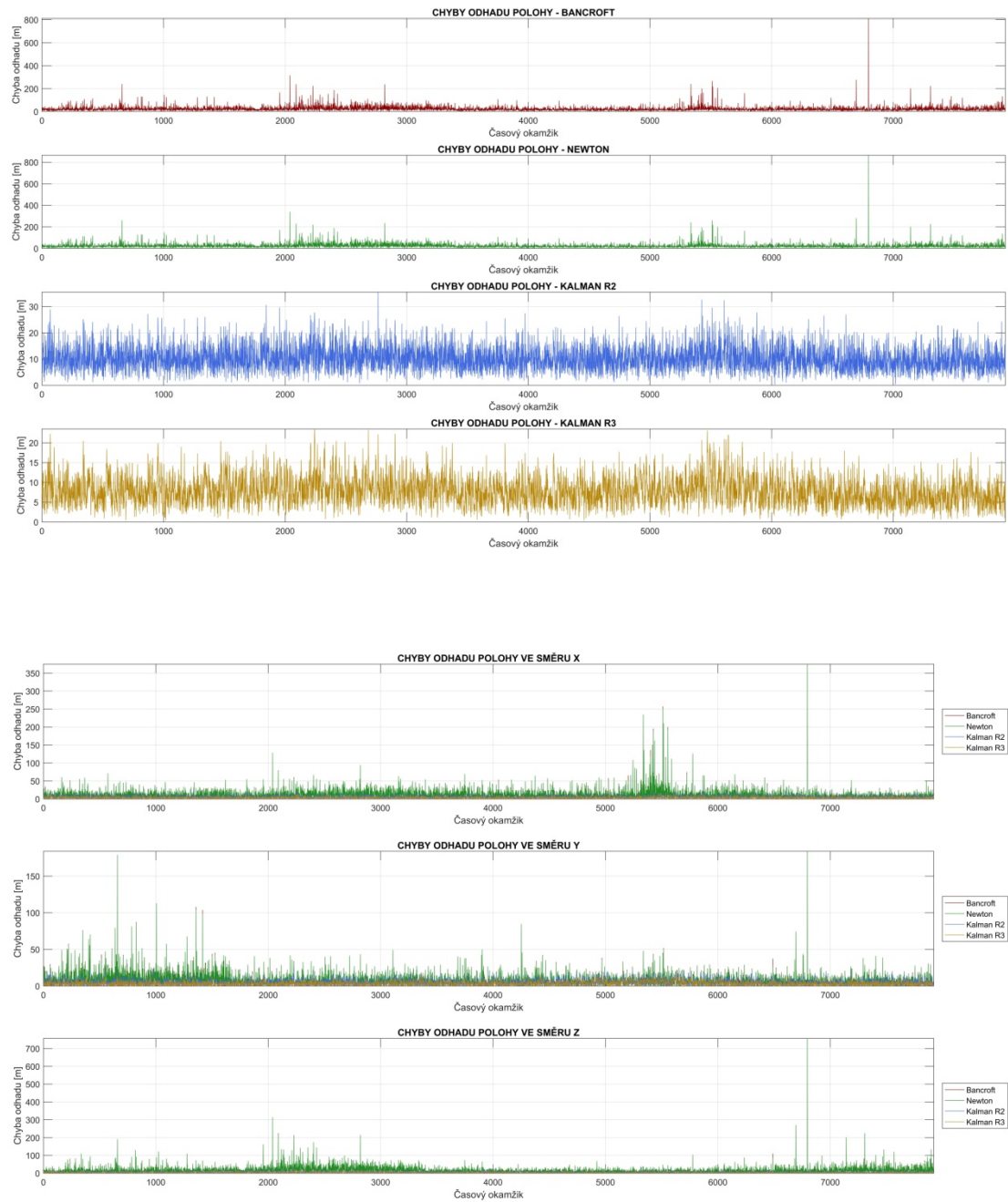


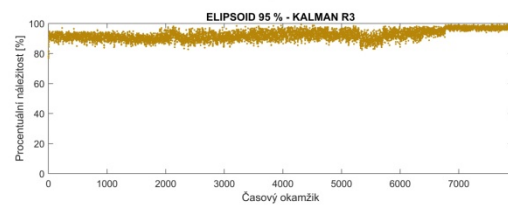
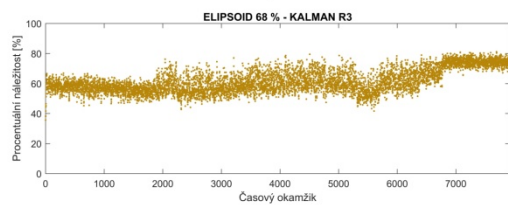
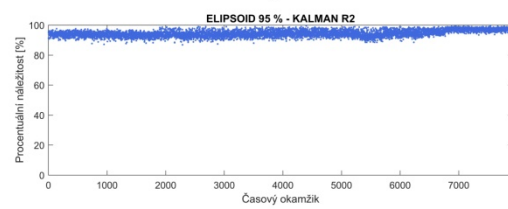
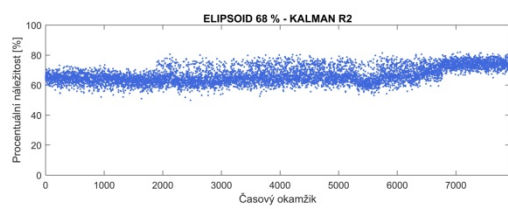
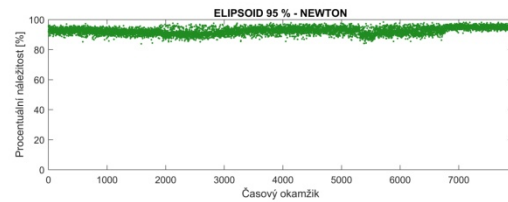
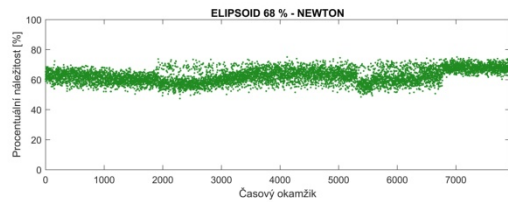
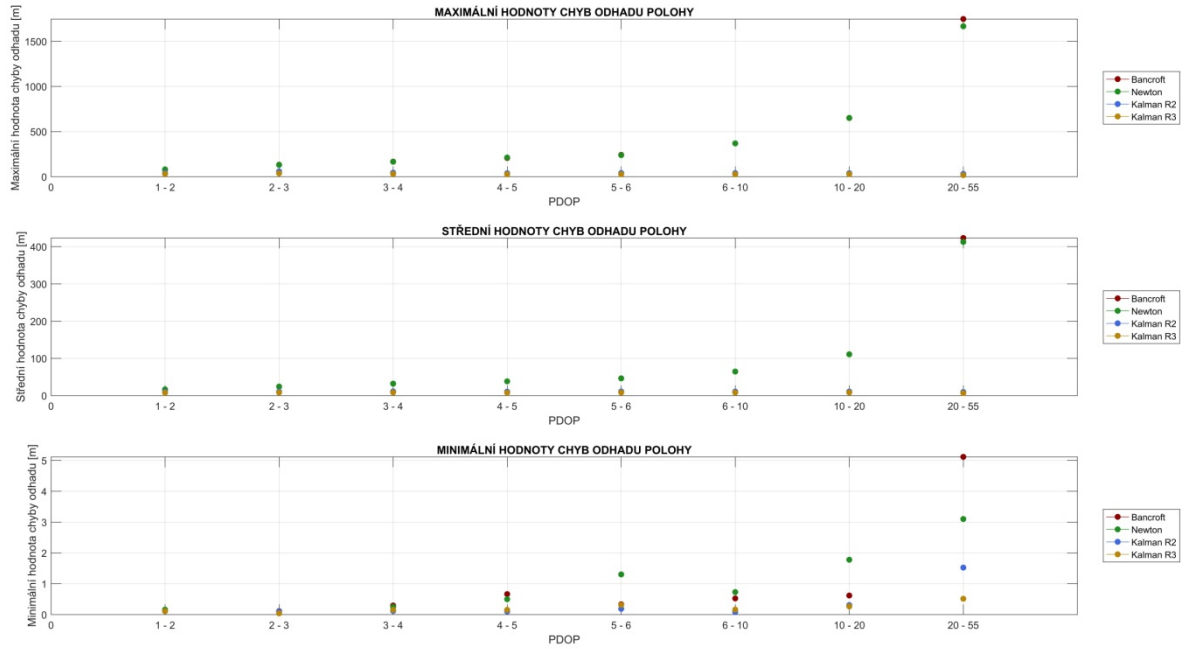


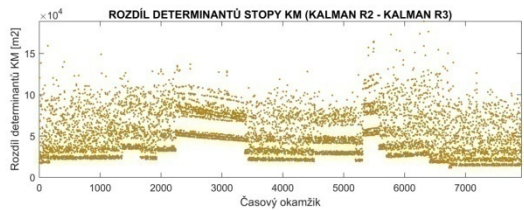
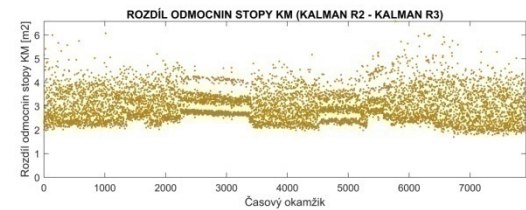
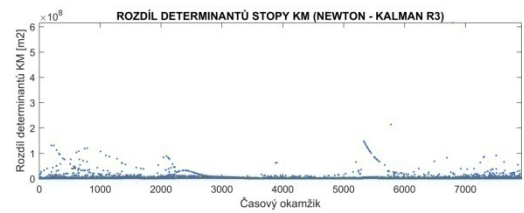
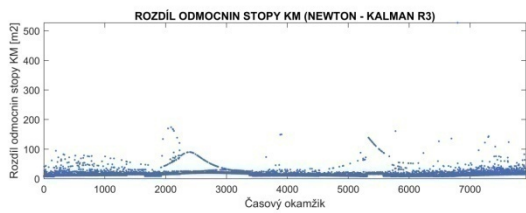
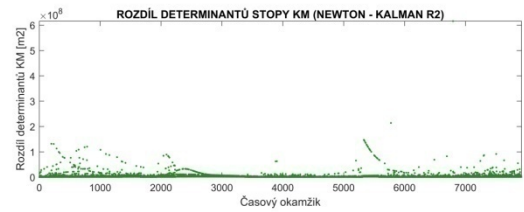
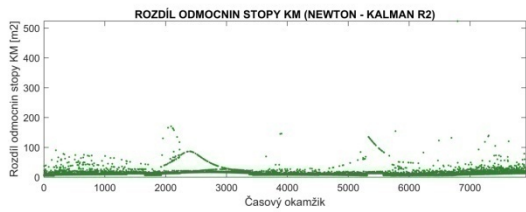
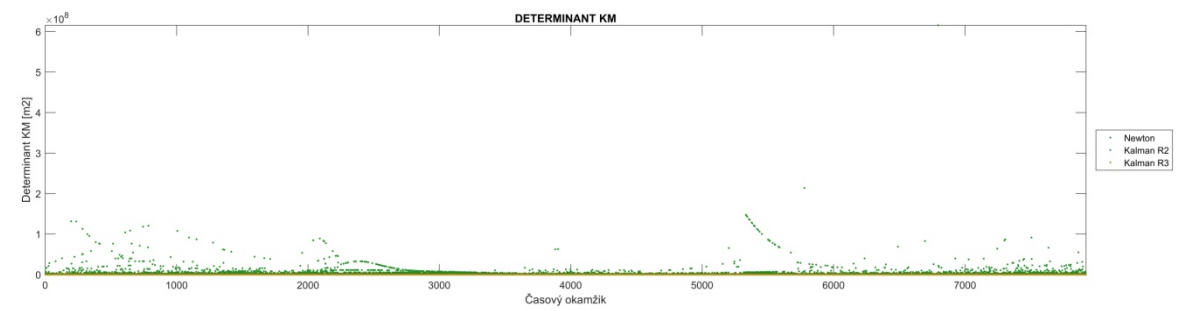
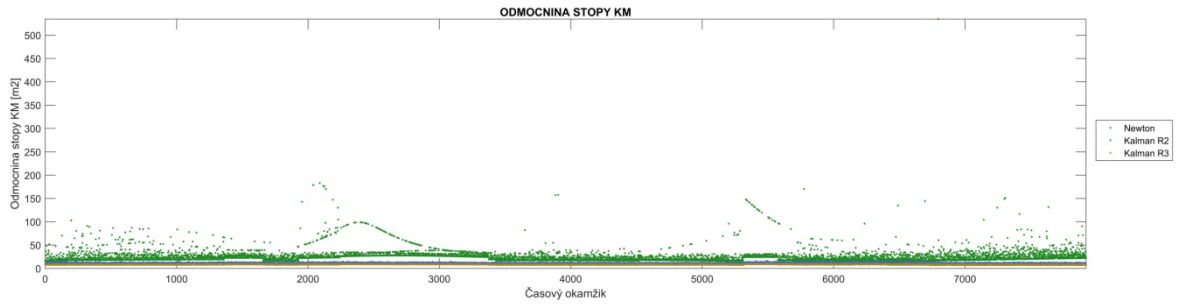


## Simulace 6



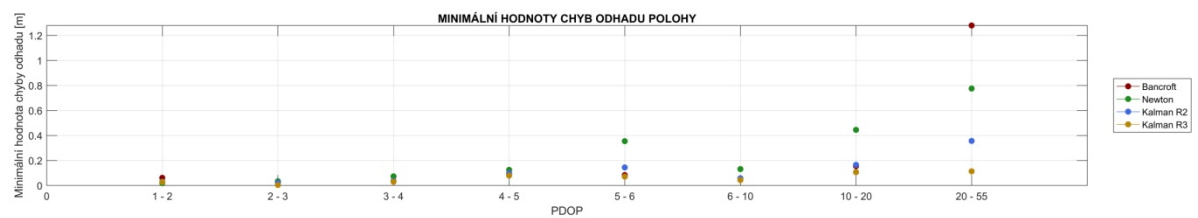
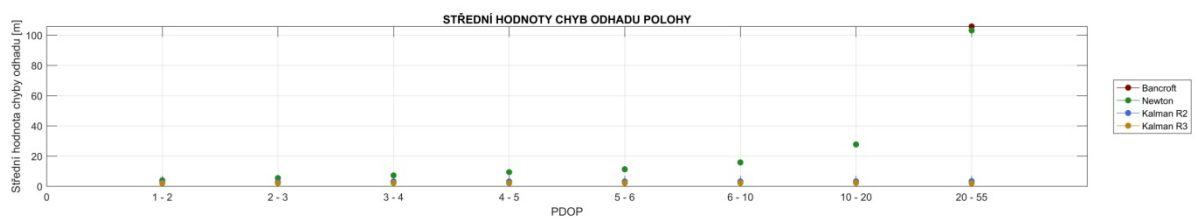
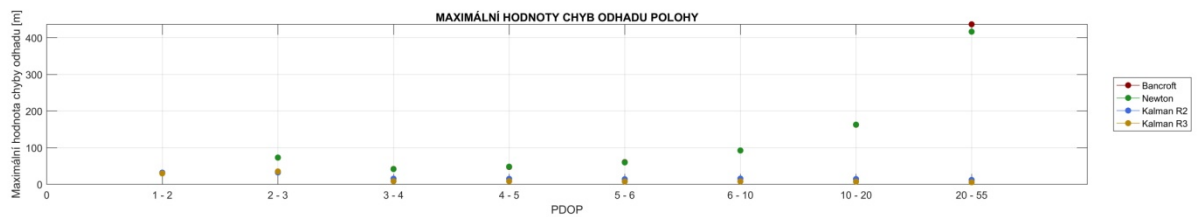
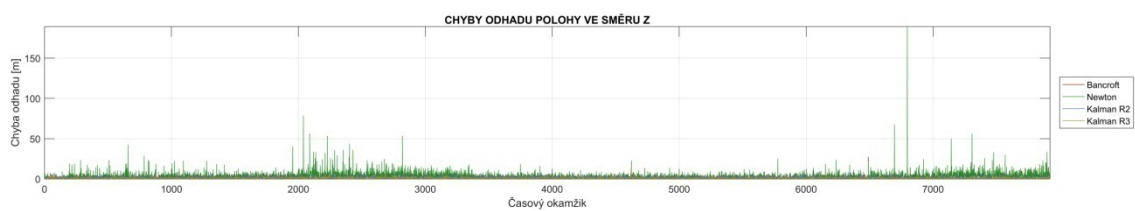
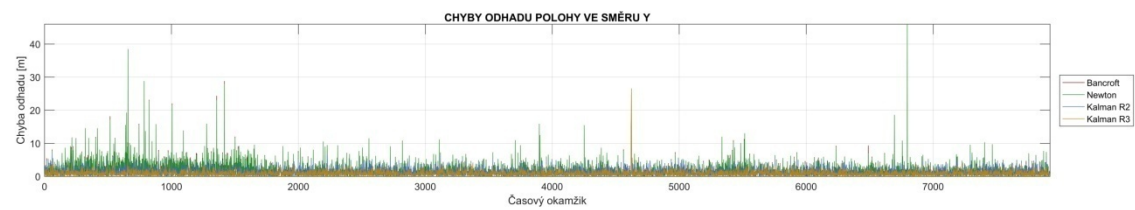
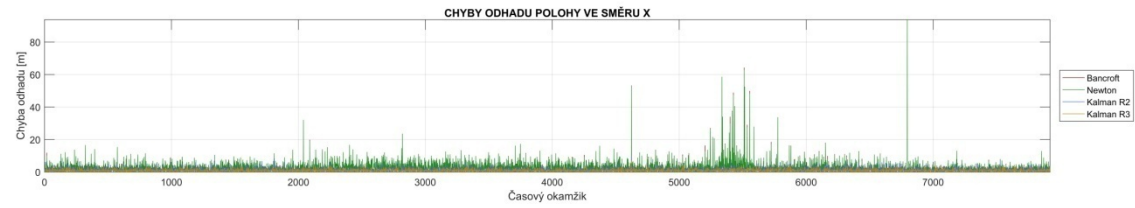


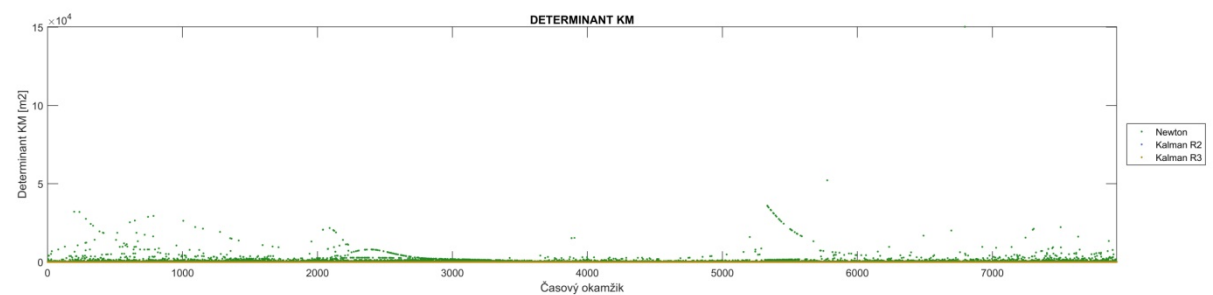
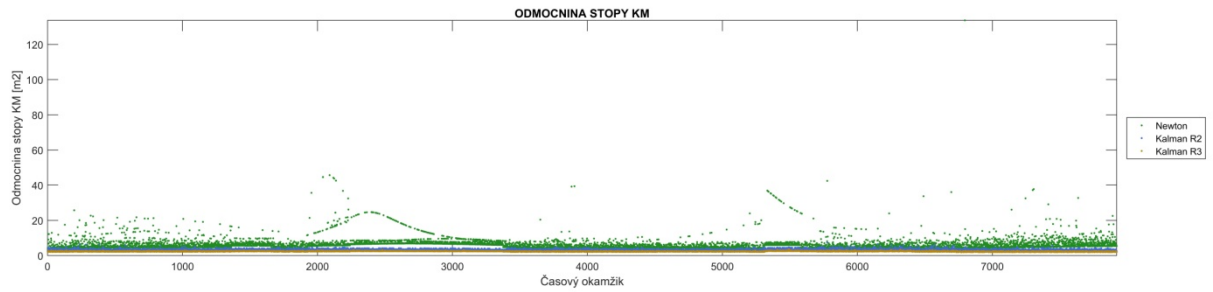
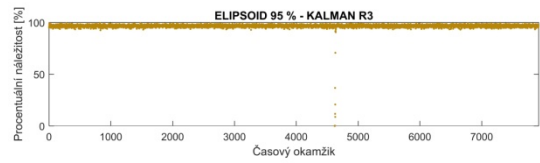
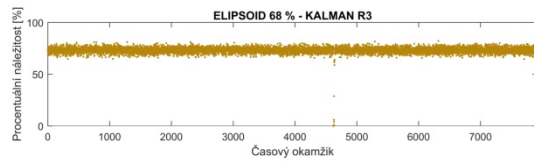
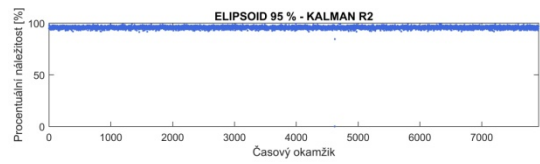
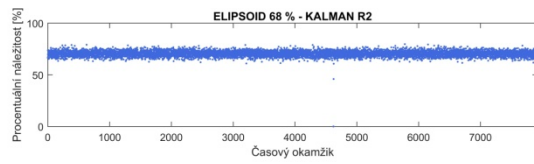
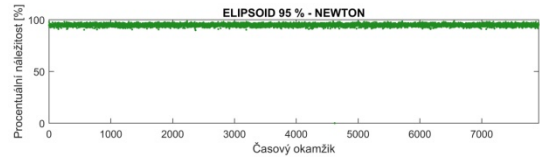
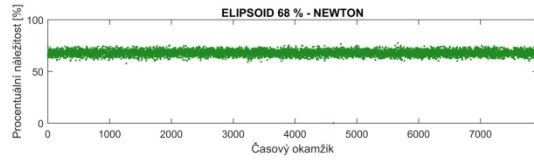


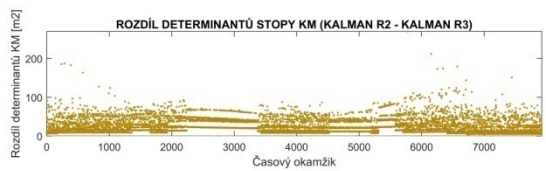
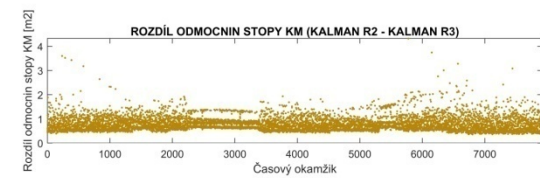
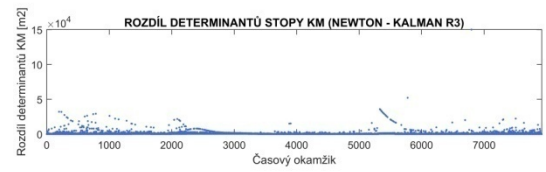
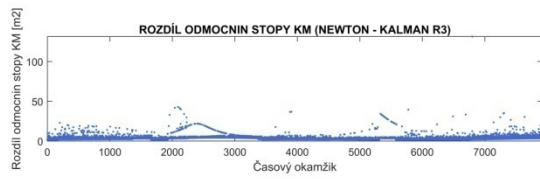
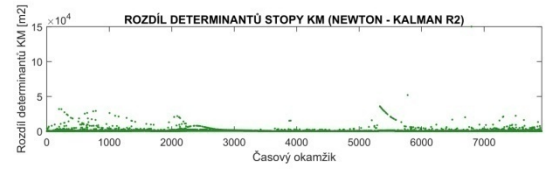
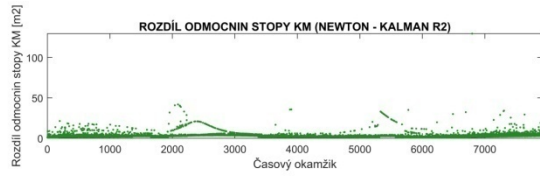


## Simulace 7

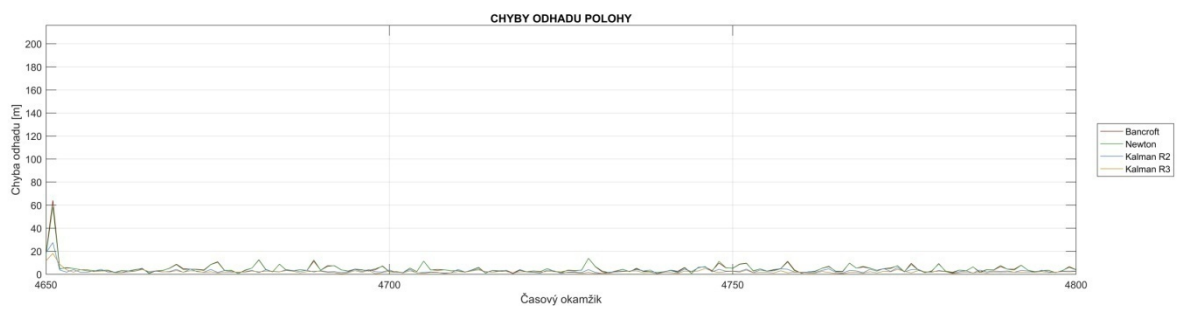
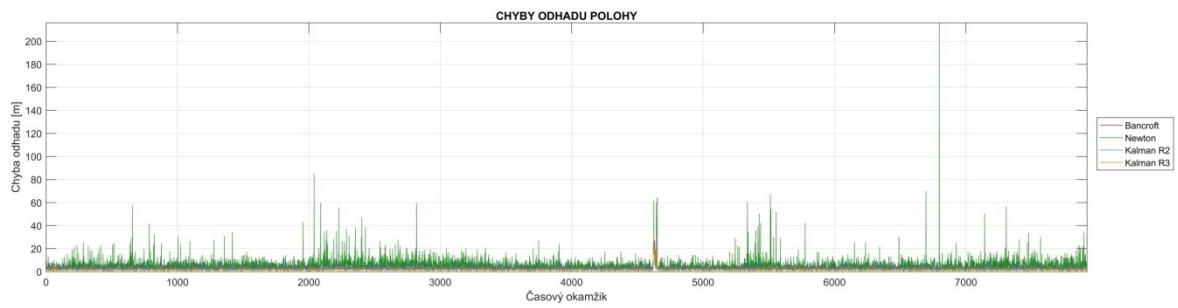




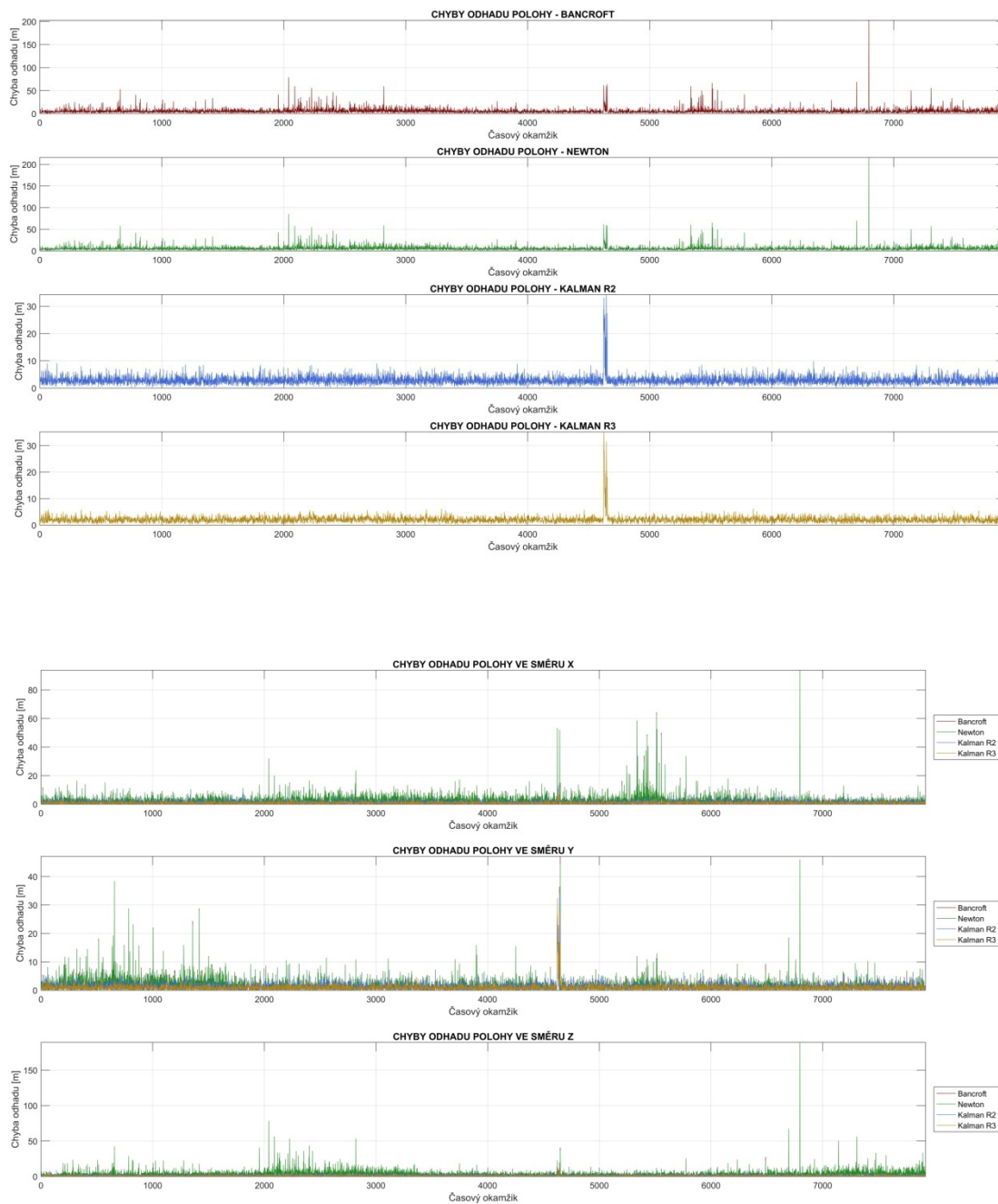


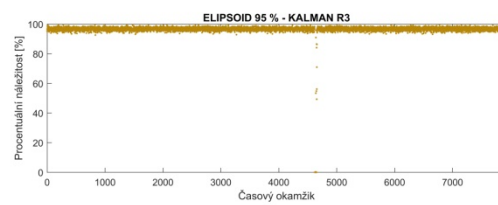
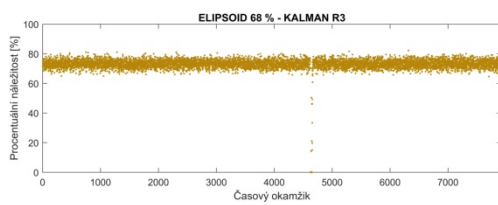
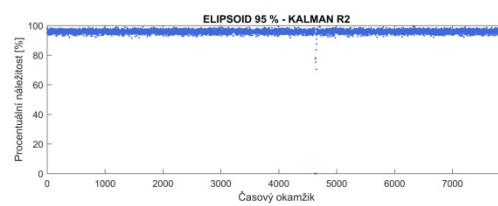
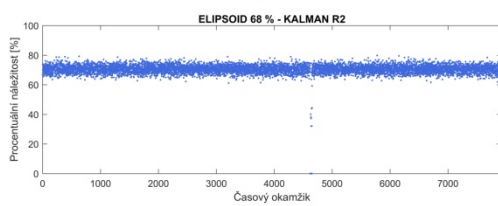
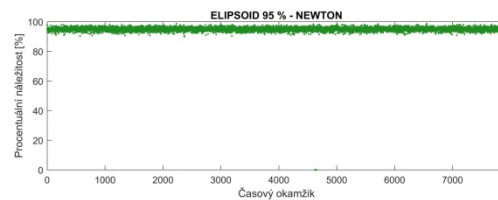
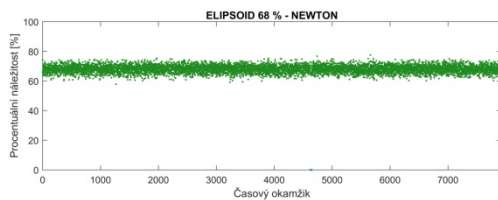
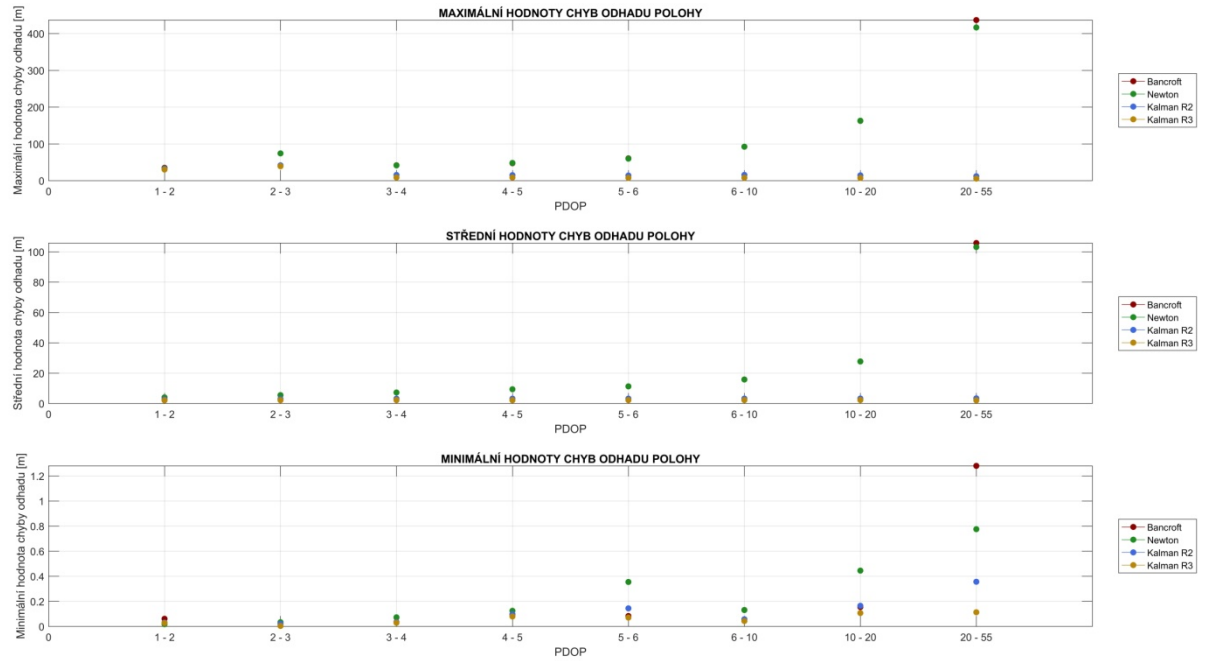


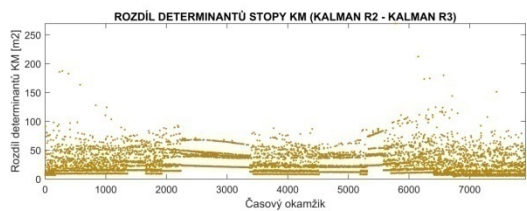
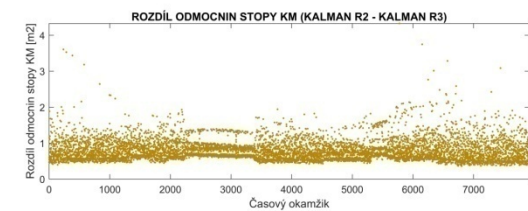
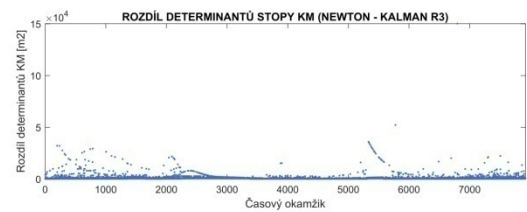
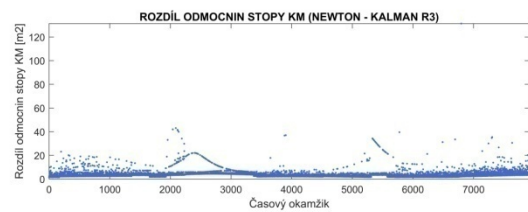
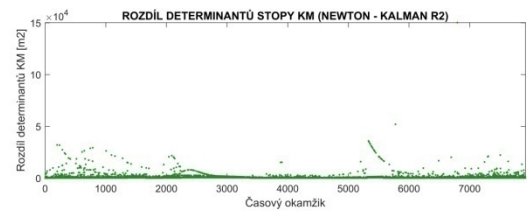
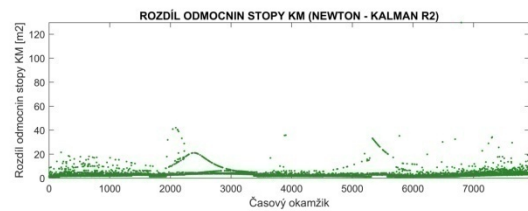
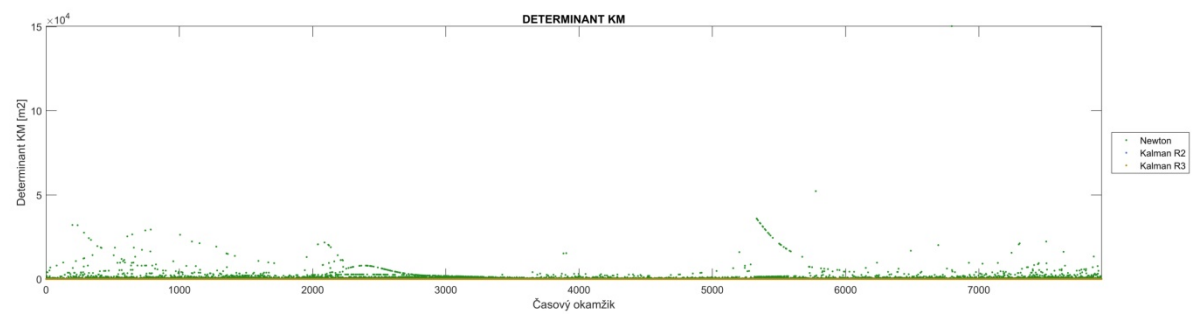
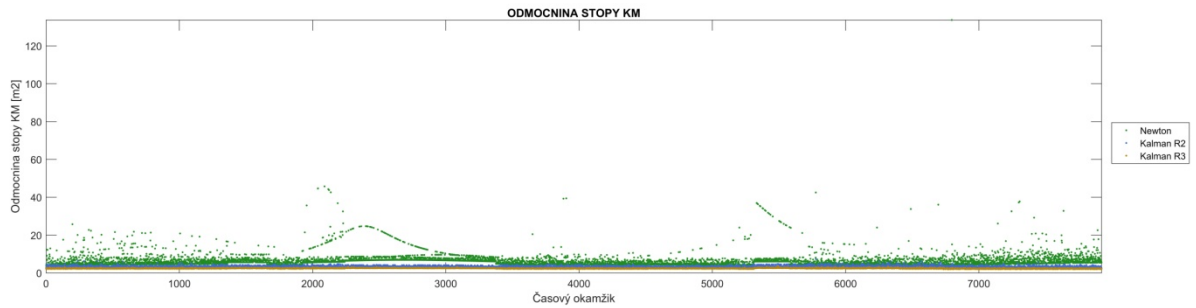
## Simulace 8











## Simulace 9

