

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická
Katedra měření

BAKALÁŘSKÁ PRÁCE



Martin Paclík

Optimalizace softwaru pro vyhodnocení navigačních dat

Vedoucí práce: doc. Ing. Jan Roháč, Ph.D.

Studijní program: Otevřená informatika

Obor: Počítačové systémy

Praha 2016



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Martin Paclík
Studijní program: Obor:	Otevřená informatika Počítačové systémy
Název tématu česky:	Optimalizace softwaru pro vyhodnocení navigačních dat
Název tématu anglicky:	Optimization of Software Dedicated for Navigation Data Processing

Pokyny pro vypracování:

Seznamte se s aktuální podobou hardwaru inerciálního navigačního systému (INS) skládajícího se z inerciální měřicí jednotky (IMU) MPU-9150 od firmy InvenSense a GNSS modulu MAX-M8 od firmy uBlox. Ke zpracování dat z dané IMU jednotky a GNSS modulu je využít mikrokontrolér STM32F405. Daný mikrokontrolér musí zajistit sběr dat, jejich předzpracování a následnou fúzi pomocí rozšířeného Kalmanova filtru (EKF). Odhady rychlosti, pozice a orientace, jako výstupy Kalmanova filtru, jsou poskytovány na výstupní sběrnici CAN (protokol CANaerospace) či RS232. Seznamte se s aktuální podobou implementace předzpracování a fúze dat v daném mikrokontroléru.

Navrhněte úpravu stávající podoby SW mikrokontroléru tak, aby se snížila jeho výpočetní náročnost a to pomocí úpravy základních funkcí či struktury. Porovnejte výpočetní náročnost v závislosti na implementovaném modelu (modely budou dodané zadavatelem práce) a též i na použité interpretaci datového typu proměnných. Těžiště práce spočívá v úpravě SW a analýze výpočetní náročnosti.

Seznam odborné literatury:

- [1] Groves P. D.: Principles of GNSS, inertial, and multisensor integrated navigation systems, ARTECH HOUSE, ISBN-13: 978-1-58053-255-6
- [2] Soták M., et al: Integrácia navigačných systemov. Košice 2006. ISBN 80-969619-9-3

Vedoucí bakalářské práce: doc. Ing. Jan Roháč, Ph.D.

Datum zadání bakalářské práce: 5. ledna 2016

Platnost zadání do¹: 30. září 2017

L.S.

Doc. Ing. Jan Holub, Ph.D.
vedoucí katedry

Prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 5. 1. 2016

¹ Platnost zadání je omezena na dobu tří následujících semestrů.

Prohlášení

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

V Praze dne

Podpis autora

Anotace

Bakalářská práce je zaměřena na urychlení výpočtu algoritmu Kalmanova filtru na navigační jednotce s mikroprocesorem ARM Cortex-M4. Součástí práce je popis různých variant a způsobů implementace Kalmanova filtru, návrh optimalizací stávajícího softwaru a porovnání jejich efektivity výpočtu na ARM Cortex-M4 mikroprocesoru.

Annotation

The Bachelor thesis is aimed on acceleration of computation of Kalman filter algorithm on navigation unit with ARM Cortex-M4 microprocessor. Different Kalman filter's variants and ways of implementation are described, optimizations of software are suggested and their efficiencies on ARM Cortex-M4 microprocessor are compared in the thesis.

Poděkování

Děkuji vedoucímu práce doc. Ing. Janu Roháčovi, Ph.D. za rady a za čas, který mi věnoval. Dále děkuji své rodině za podporu při studiu.

Obsah

1	Úvod	1
2	Rozbor	2
2.1	Vztažné soustavy a souřadnicové systémy	2
2.2	Inerciální navigační systém (INS)	3
2.2.1	Inerciální měřicí jednotka (IMU)	4
2.2.2	Navigační počítač	4
2.2.3	Akcelerometry	4
2.2.4	Senzory úhlové rychlosti a gyroskopy	5
2.2.5	Výhody a nevýhody INS	7
2.3	Globální navigační satelitní systém (GNSS)	7
2.3.1	Globální poziční systém (GPS)	7
2.4	Kalmanův filtr	8
2.4.1	Rozšířený Kalmanův filtr (EKF)	9
2.4.2	Josephův tvar korekce kovarianční matice	11
2.4.3	Výpočet Kalmanova zesílení pomocí UD rozkladu	11
2.4.4	Sekvenční korekce	12
2.4.5	Bierman-Joseph	13
2.4.6	De Vries-Joseph	13
3	Navigační jednotka	15
3.1	Hardware	15
3.2	Neoptimalizovaný výpočet EKF	16
4	Optimalizace	18
4.1	ARM CMSIS DSP Library	18
4.2	Korekční krok EKF	18
4.2.1	Implementace	19
4.3	Predikce \mathbf{P}_k pomocí operací se submaticemi	20
4.3.1	Implementace	20
4.4	Předpočítávání	21
5	Výsledky	22
5.1	Způsob měření	22
5.2	Korekční krok EKF	22
5.2.1	Měření	23
5.2.2	Shrnutí	27
5.3	Predikce \mathbf{P}_k pomocí operací se submaticemi	29

5.3.1	Měření	29
5.3.2	Shrnutí	30
5.4	Předpočítávání	31
6	Závěr	32
	Seznam použité literatury	33
	Seznam zkratk	34
	Seznam symbolů	35
	Příloha A CD	36
	Příloha B Naměřené hodnoty a grafy	37
B.1	Metody korekce	37
B.1.1	Standardní EKF	37
B.1.2	Inverze pomocí UD dekompozice	38
B.1.3	Sekvenční korekce	39
B.1.4	De Vries-Joseph s UD dekompozicí	40
B.1.5	De Vries-Joseph, sekvenční	41
B.1.6	Bierman-Joseph	42
B.1.7	Metoda s nejmenší časovou náročností (bez ARM DSP Library) . . .	43
B.1.8	Metoda s nejmenší časovou náročností (s ARM DSP Library)	44
B.2	Metody predikce	45
B.2.1	Klasické operace vs. operace se submaticemi (bez ARM DSP Library) . . .	45
B.2.2	Klasické operace vs. operace se submaticemi (s ARM DSP Library) . . .	46

1 Úvod

Dějiny navigace jsou asi stejně dlouhé jako lidstvo samo. Člověk vždy potřeboval umět odhadnout svou polohu a rychlost, resp. polohu a rychlost jiných objektů, a využíval k tomu množství nejrůznějších metod.

V dnešní době patří mezi nejvýznamnější systémy pro určení polohy globální navigační satelitní systém (GNSS) a inerciální navigační systém (INS). Vedle nich existuje celá řada senzorů a systémů poskytující dílčí navigační informace. Například barometr pro určování výšky či kompas pro určení orientace vůči světovým stranám.

Žádný z těchto systémů a senzorů nám neposkytne dokonalou informaci o poloze či rychlosti, ale pouze odhad zatížený chybou. Odhady z více systémů lze různě kombinovat s cílem získat co nejpřesnější konečný odhad a optimálně potlačit měřený šum. Jednou z nejpoužívanějších metod pro kombinaci více odhadů je dnes Kalmanův filtr.

Kalmanův filtr je výpočetně náročný. Při výpočtu se operuje s maticemi a ve své standardní podobě vyžaduje spočítání inverzní matice, což je jeho časově nejnáročnější část. Často je potřeba, aby výpočet probíhal v reálném čase s vysokou frekvencí. Existují proto různé varianty a způsoby implementace Kalmanova filtru, které snižují výpočetní náročnost, a to často buď efektivním výpočtem inverzní matice anebo obejitím nutnosti jejího výpočtu.

Cílem této bakalářské práce je urychlit výpočet Kalmanova filtru na navigační jednotce s mikroprocesorem ARM Cortex-M4 úpravou jejího stávajícího softwaru. S tím souvisí i porovnání efektivity výpočtu různých variant a způsobů implementací Kalmanova filtru na tomto mikroprocesoru.

2 Rozbor

Tato kapitola se věnuje teoretickému rozboru problematiky inerciálních navigačních systémů (INS), globálních navigačních satelitních systémů (GNSS) a především pak způsobům jejich integrace pomocí Kalmanova filtru. Nejdříve jsou popsány různé vztažné soustavy a souřadnicové systémy používané v navigaci, dále INS a GNSS a nakonec je rozebrán Kalmanův filtr a jeho různé varianty a způsoby implementace.

2.1 Vztažné soustavy a souřadnicové systémy

V této části jsou stručně popsány některé souřadnicové systémy a s nimi svázané vztažné soustavy, se kterými je možné se v navigaci setkat. Podrobnější popis je v [8] a [1].

Vztažná soustava je fyzikální objekt s jednoznačně určenou polohou a orientací, vůči kterému sledujeme pohyb jiných objektů. Aby bylo možné tento pohyb popsat, je třeba zavádět souřadnicové systémy. Každá vztažná soustava má nekonečně mnoho souřadnicových systémů. Dále jsou uvedeny ty nejpoužívanější. Jejich znázornění je na obrázku 2.1.

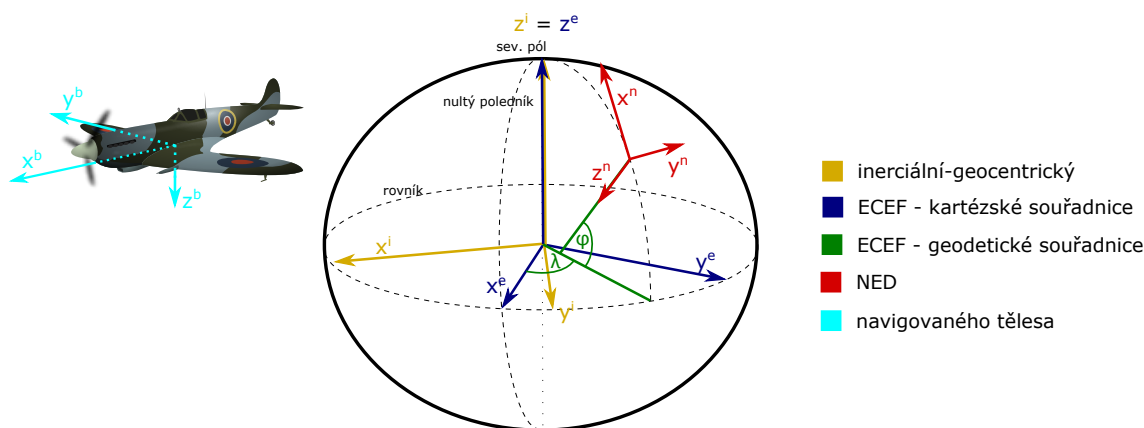
Earth-centered inertial (ECI)

ECI je souřadnicový systém svázaný s inerciální geocentrickou vztažnou soustavou. Inerciální geocentrickou vztažnou soustavu tvoří Země, u níž se nebere v potaz její rotace. ECI má počátek v hmotném středu Země. Osa z je totožná s osou Zemské rotace se směrem k severnímu pólu. Osa x směřuje k souhvězdí Berana a osa y dotváří pravotočivý souřadnicový systém.

Earth centered, Earth fixed (ECEF)

ECEF je jedním z možných souřadnicových systémů zemské vztažné soustavy. Zemská vztažná soustava se liší od inerciální-geocentrické tím, že se bere v úvahu rotace Země kolem její osy. ECEF má počátek v hmotném středu Země. Osa z je totožná s osou z inerciálního souřadnicového systému. Osa x protíná průsečík nultého poledníku a rovníku. Osa y dotváří pravotočivý souřadnicový systém. ECEF rotuje společně se Zemí a jednou za 24 hodin je totožný s inerciálním souřadnicovým systémem.

Polohu v ECEF souřadnicovém systému lze kromě karteziánských souřadnic popsat i souřadnicemi geodetickými, tedy zeměpisnou šířkou φ , zeměpisnou délkou λ a výškou h nad referenčním elipsoidem. Při popisu polohy geodetickými souřadnicemi je tedy nutné mít definovaný referenční elipsoid, který co nejlépe odpovídá tvaru Země. Jednou z možností je standard WGS 84, který je používán systémem GPS. Zeměpisná délka je úhel v rozsahu -180° až 180° , který svírá rovina nultého poledníku s rovinou poledníku, který

Obrázek 2.1: Souřadnicové systémy podle [1] ¹

prochází popisovaným bodem. Zeměpisná šířka je určena úhlem v rozmezí -90° až 90° mezi rovinou rovníku a normálou referenčního elipsoidu v popisovaném bodě.

North-East-Down (NED)

NED je jedním z lokálních navigačních souřadnicových systémů spjatých s povrchem Země. Počátek souřadnic je umístěn na povrchu Země a je definován geodetickými souřadnicemi. Osa x směřuje na sever, y na východ a z ve směru Zemské tíže.

Souřadnicový systém navigovaného tělesa

Navigované těleso tvoří vztažnou soustavu (*body frame*). S ní souvisící souřadnicový systém navigovaného tělesa (*body coordinate system*) je definován takto: počátek souřadnic je v těžišti dopravního prostředku, osa x leží v rovině symetrie a směřuje kupředu, osa y je na rovinu symetrie kolmá a směřuje doprava, a osa z dotváří pravotočivý souřadnicový systém.

2.2 Inerciální navigační systém (INS)

Funkcí INS je určování pozice, rychlosti a orientace objektu bez vnějších signálů pouze za pomoci inerciálních senzorů integrací měřeného zrychlení na rychlost, měřené úhlové rychlosti na úhly natočení a dvojitou integrací měřeného zrychlení na pozici. Hlavními součástmi inerciálního navigačního systému, jsou inerciální měřicí jednotka (IMU) a navigační počítač. INS může být dvojího druhu [8]:

1. **strapdown**: inerciální senzory jsou spojeny s navigovaným tělesem tak, že jejich osy citlivosti mají stejný směr jako hlavní osy navigovaného tělesa,
2. **s pohyblivou základnou**: inerciální senzory jsou připevněny k plošině, jejíž orientace vůči inerciální vztažné soustavě je udržována jako konstantní.

Text dále se vztahuje ke strapdown INS.

¹Obrázek letadla (volné dílo) převzat z: <http://publicdomainvectors.org/cs/volnych-vektoru/Supermarine-Spitfire-letadlo-vektorove-ilustrace/12140.html>

2.2.1 Inerciální měřicí jednotka (IMU)

IMU se skládá ze soustavy inerciálních senzorů. Hlavními senzory v IMU jsou akcelerometry a senzory úhlové rychlosti (popsány níže). Jak akcelerometry, tak senzory úhlové rychlosti měří vzhledem k inerciální vztahné soustavě. Většinou se používají tříosé varianty měřicí ve třech ortogonálních směrech, které se zpravidla umísťují tak, aby tyto směry byly shodné s orientací os souřadnicové soustavy navigovaného tělesa. Pak jsou naměřené hodnoty vyjádřeny v souřadnicovém systému navigovaného tělesa.

2.2.2 Navigační počítač

Poté, co IMU naměří specifickou sílu a úhlovou rychlost, navigační počítač spočítá tzv. navigační rovnice, jejichž výsledkem jsou derivace polohy a rychlosti objektu v referenčním souřadnicovém systému. Ty pak navigační počítač integruje do odhadů aktuální polohy a rychlosti. Navigační rovnice v různých referenčních souřadnicových systémech jsou popsány v [8].

2.2.3 Akcelerometry

Akcelerometry měří specifickou sílu \mathbf{f}_{ib}^b . Akcelerometr ležící na povrchu Země naměří specifickou sílu rovnou $-\mathbf{g}$. Ve volném pádu naměří specifickou sílu nulovou. Platí, že

$$\mathbf{f}_{ib} = \mathbf{a}_{ib} - \mathbf{g},$$

kde \mathbf{a}_{ib} je zrychlení způsobené změnou rychlosti pohybu a \mathbf{g} je gravitační zrychlení.

Různé druhy akcelerometrů jsou popsány v [8]. Mezi základní patří:

- mechanické,
- vibrační,
- s pevnou strukturou,
- MEMS (piezoelektrické / piezorezistivní / kapacitní).

Stále více jsou používány akcelerometry vyrobené technologií MEMS.

MEMS (*Micro-Electro-Mechanical Systems*) Jedná se o technologii, pomocí které se vytváří miniaturní mechanické systémy v řádu mikrometrů na křemíkovém podkladu. Využívá se postupů vzniklých pro výrobu integrovaných obvodů. Tyto postupy jsou stále zdokonalovány a lze jimi vyrábět čím dál tím menší systémy. Spolu se zdokonalováním výrobních procesů jsou zdokonalovány i vlastnosti MEMS senzorů.

Kapacitní akcelerometry

Kapacitní akcelerometr obsahuje uvnitř hmotu, která je uchycena tak, aby mohla v jedné ose (v ose citlivosti) měnit svou pozici vůči pouzdru senzoru. Působí-li na senzor

v ose citlivosti zrychlení, hmota má tendenci setrvat v klidu vůči inerciální vztažné soustavě na rozdíl od pouzdra senzoru. Výsledkem je změna pozice hmoty vůči pouzdru v ose citlivosti. K vyhodnocení této změny se u kapacitních akcelerometrů využívají diferenční kondenzátory. Dvě elektrody jsou pevně spjaty s pouzdrum senzoru a jedna (prostřední) je uchycena ke hmotě. Dohromady tak elektrody tvoří dva kondenzátory. Když na senzor působí zrychlení, prostřední elektroda mění svou vzdálenost vůči ostatním a tím se mění i kapacita kondenzátorů, která je na vzdálenosti elektrod závislá podle vztahu

$$C = \varepsilon \frac{S}{d},$$

kde ε je permitivita dielektrika mezi elektrodami, S je plocha elektrod a d je vzdálenost mezi elektrodami.

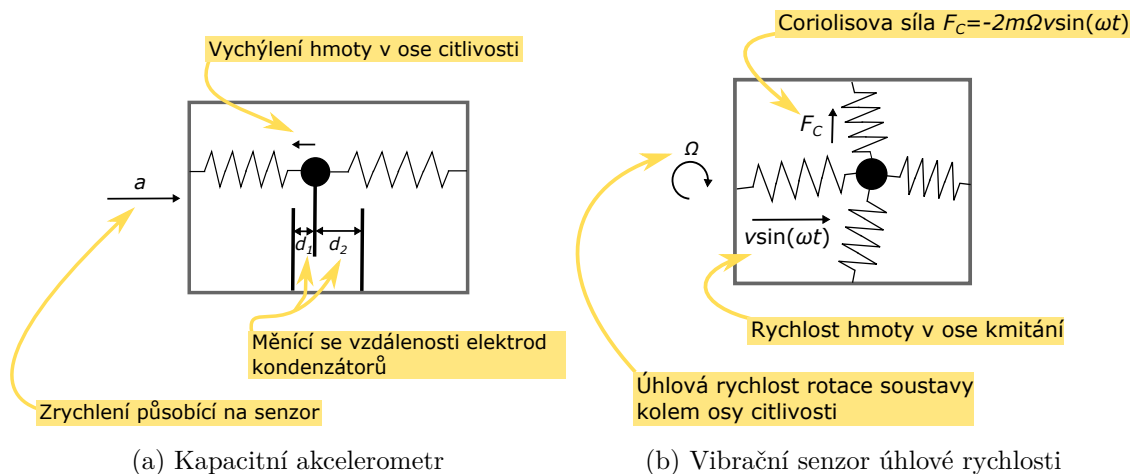
Kapacitní akcelerometry jsou často vyráběny technologií MEMS. Princip kapacitního akcelerometru je znázorněn na obrázku 2.2a. Podrobněji je tento typ senzoru popsán v [8].

2.2.4 Senzory úhlové rychlosti a gyroskopy

Gyroskopy a senzory úhlové rychlosti jsou zařízení pro měření úhlu, případně úhlové rychlosti ω_{ib}^b . Je známo více principů, na jejichž základě mohou gyroskopy a senzory úhlové rychlosti pracovat. Různé druhy jsou popsány v [8]. Mezi základní druhy patří:

- mechanické,
- vibrační (rezonanční),
- optické
 - s pevnou optickou základnou (RLG - *Ring Laser Gyro*),
 - vláknové (FOG - *Fiber Optic Gyro*).

Mezi nejpresnější a zároveň nejdražší patří RLG. Využívají se například ve velkých dopravních letadlech.



Obrázek 2.2: Principy inerciálních senzorů

Vibrační senzory úhlové rychlosti

Vibrační typ senzoru úhlové rychlosti je většinou vyráběn pomocí MEMS technologie a patří mezi všeobecně nepoužívanější díky nízké ceně. Se zdokonalováním MEMS technologie a zlepšováním parametrů MEMS senzorů nachází vibrační MEMS senzory úhlové rychlosti stále větší uplatnění i v levných INS. Jejich funkce je založena na Coriolisově síle. Princip vibračního senzoru úhlové rychlosti je znázorněn na obrázku 2.2b.

Coriolisova síla Pohybuje-li se těleso v rotující vztažné soustavě, pozorovateli pohybujícímu se společně s touto soustavou se zdá, že na těleso působí síla odchylojící ho od přímé trajektorie, ačkoli vůči inerciální vztažné soustavě těleso koná rovnoměrný přímočarý pohyb a žádná síla na něj ve skutečnosti nepůsobí. Takovéto zdánlivé síle způsobené neinerciálností soustavy se říká Coriolisova.

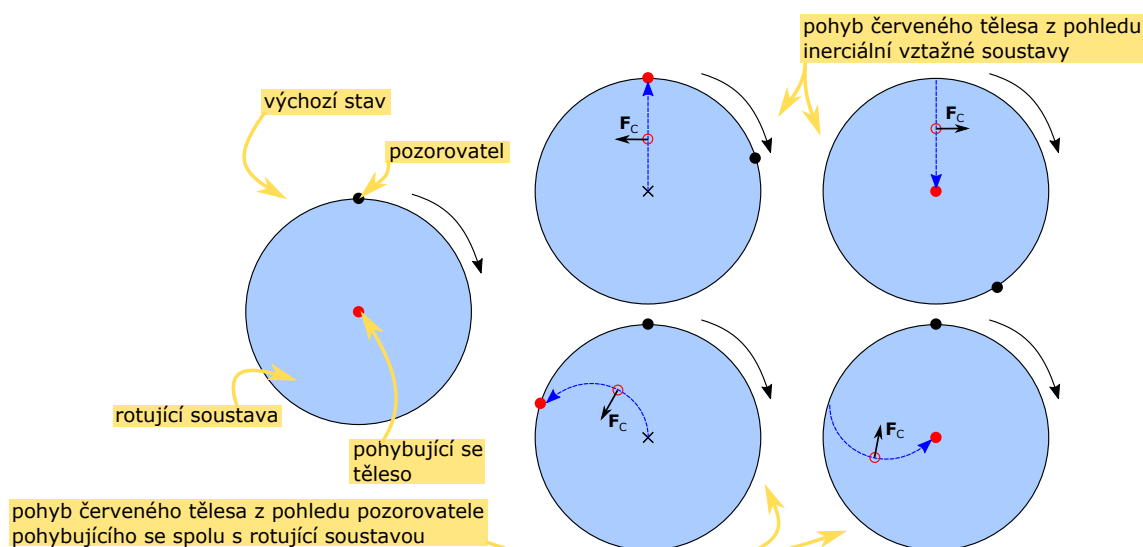
Coriolisova síla se spočítá jako

$$\mathbf{F}_C = -2m\boldsymbol{\omega} \times \mathbf{v},$$

kde $\boldsymbol{\omega}$ je úhlová rychlost otáčení soustavy, \mathbf{v} je rychlost pohybujícího se tělesa a m je jeho hmotnost. Působení Coriolisovy síly je znázorněno na obrázku 2.3.

Vibrační sensor úhlové rychlosti obsahuje hmotu kmitající v jedné ose (osa kmitání). Zároveň umožňuje této hmotě vychýlení v jedné ose ortogonální (osa vyhodnocení). Třetí osa, která je kolmá na obě osy předešlé, je tzv. osa citlivosti. Rotuje-li sensor kolem osy citlivosti a hmota v senzoru kmitá v ose kmitání, pak je možné detekovat změnu polohy hmoty vůči pouzdru v ose vyhodnocení díky působení Coriolisovy síly.

Podrobněji je tento typ senzoru popsán v [8].



Obrázek 2.3: Coriolisova síla

2.2.5 Výhody a nevýhody INS

Žádné měření nemůže být zcela přesné kvůli nedokonalosti senzorů. Senzor vždy změří součet skutečné hodnoty a chyby. Hlavními chybami senzorů jsou nelinearita, bias (systematická chyba přičtená ke každému měření) a náhodný šum. Čím lepší je senzor, tím menší tyto chyby jsou, nikdy je však nejde úplně odstranit.

Navigační počítač integrující měřené zrychlení a úhlovou rychlost tedy kromě skutečného zrychlení a úhlové rychlosti integruje do rychlosti a polohy také změřené chyby. Chyba v určené poloze a rychlosti pomocí INS je časově závislá - čím déle INS pracuje, tím více bylo integrováno chyb do odhadů a tím méně lze věřit, že odhady odpovídají skutečnosti.

Nekonečnému nárůstu chyby lze zabránit například tím, že jsou odhady INS čas od času vhodně korigovány na základě měření různých podpůrných senzorů a systémů. Může to být například barometr pro měření výšky, senzor rychlosti, globální navigační satelitní systém atd. Takový INS se nazývá aided INS (AINS).

Výhodou INS je nezávislost na vnějších signálech a vyšší obnovovací frekvence oproti GNSS (viz dále).

2.3 Globální navigační satelitní systém (GNSS)

GNSS (*Global Navigation Satellite System*) je systém pro určování polohy, rychlosti a času. Satelity na oběžné dráze Země vysílají signál, na jehož základě přijímače počítají navigační informace. V současnosti fungují, nebo jsou ve vývoji, následující systémy:

- **GPS** je spravovaný Ministerstvem obrany USA. Primárně se jedná o vojenský systém, který je nyní přístupný i v civilním sektoru. Více je popsán níže.
- **GLONASS** je obdobný ruský systém.
- **Galileo** je vyvíjený Evropskou kosmickou agenturou. Na rozdíl od předchozích dvou se jedná o civilní systém. Plné funkčnosti, při které bude k dispozici 30 satelitů, by měl dosáhnout do roku 2020.
- **BeiDou** je čínský systém, který by měl rovněž být globálně plně funkční do roku 2020. Zatím nabízí služby uživatelům v asijsko-pacifickém regionu.

2.3.1 Globální poziční systém (GPS)

GPS (*Global Positioning System*) je navigační systém USA. Skládá se ze tří segmentů. První z nich je segment kosmický, tedy satelity na oběžné dráze. Druhý je segment řídicí, který má na starost kontrolu satelitů a jejich údržbu, jako je korekce a synchronizace hodin. Třetí je segment uživatelský zahrnující všechny přijímače využívající GPS signál.

Satelity vysílají zprávy, ze kterých lze určit jejich polohu a přesný čas, kdy byla zpráva vyslána. Přijímač zná čas, kdy zprávu přijal, a může tedy spočítat vzdálenost od satelitu. Hodiny přijímačů jsou ale nepřesné a vždy obsahují bias, se kterým se musí počítat. Proto pro přesné určení 3D polohy je nutné, aby byly v dosahu neméně 4 satelity pro konstrukci alespoň čtyř rovnic o čtyřech neznámých (3 souřadnicích a neznámém biasu hodin přijímače).

Poloha je poskytována nejčastěji v ECEF systému v geodetických souřadnicích s referenčním elipsoidem podle WGS 84.

WGS 84 (*World Geodetic System 1984*) Jedná se geodetický standard z roku 1984 definující souřadnicový systém, referenční elipsoid a geoid (ekvipotenciála gravitace) pro Zemi. V tabulce 2.1 jsou 4 parametry definované standardem WGS 84 podle [6].

Parametr	Značení	Hodnota	Jednotka
Hlavní poloosa referenčního elipsoidu	a	6378137	metry
Převrácená hodnota zploštění Země	$1/f$	298,257223563	
Úhlová rychlost otáčení Země	ω	7292115×10^{-11}	$\text{rad}\cdot\text{s}^{-1}$
Gravitační parametr	GM	$3986004,418 \times 10^8$	m^3s^{-2}

Tabulka 2.1: Parametry definované WGS 84

Z těchto definovaných parametrů lze další dopočítat. Například vedlejší poloosa b referenčního elipsoidu lze spočítat díky vztahu

$$f = \frac{a - b}{a}$$

jako $b \approx 6356752,3142$ metrů.

Výhodou GPS (i jiného GNSS) je, že přesnost odhadu navigačních informací nezávisí na čase. Nevýhodou je závislost na vnějším signálu a nižší obnovovací frekvence oproti INS. Obnovovací frekvence GPS přijímačů se zpravidla pohybuje okolo 1 Hz až 20 Hz, nejvýkonnější přijímače dosahují 100 Hz [7].

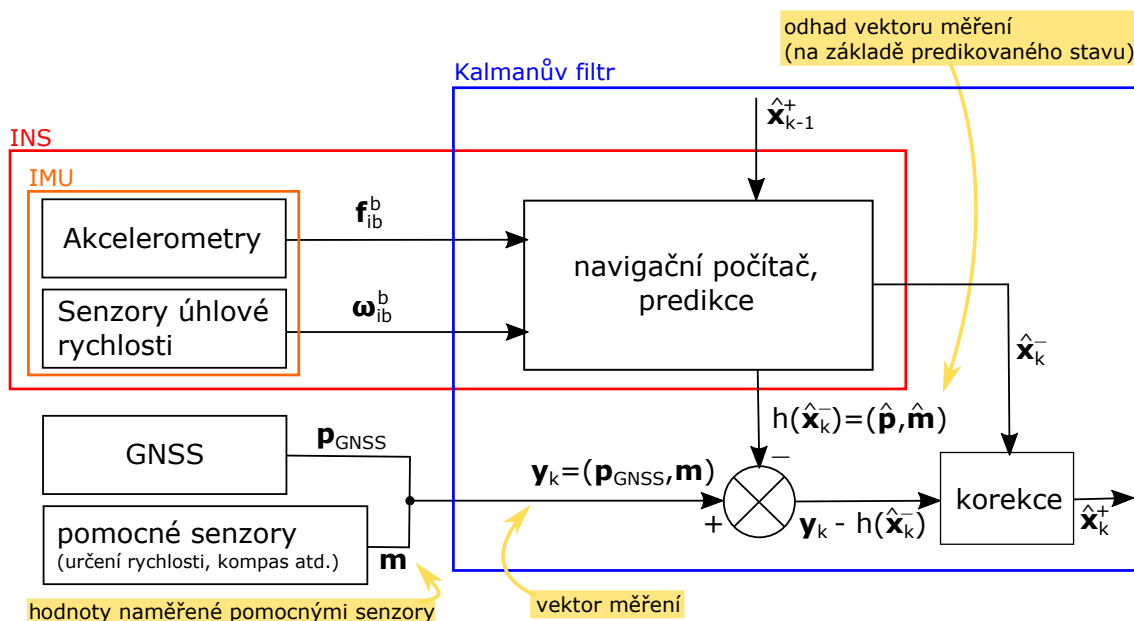
2.4 Kalmanův filtr

INS a GPS (nebo i jiný GNSS) se vhodně doplňují. Jejich vhodnou integrací vznikne systém, který

1. poskytuje navigační informace s vyšší obnovovací frekvencí než samotný GPS,
2. na rozdíl od samotného INS neztrácí zcela přesnost při dlouhodobém měření,
3. může fungovat i po ztrátě GPS signálu.

Takový systém se nazývá INS/GPS integrovaný nebo GPS-aided INS systém.

Nejčastější způsob integrace GPS a INS je za pomoci Kalmanova filtru. Pro zpřesnění odhadů navigačního systému se dále můžou použít nejrůznější další podpůrné senzory a systémy poskytující dílčí navigační informace (výškoměry, kompasů atd.). Schéma integrace INS, GNSS a dalších pomocných sensorů a systémů je na obrázku 2.4.



Obrázek 2.4: Integrace INS, GNSS a pomocných senzorů pomocí Kalmanova filtru

Kalmanův filtr slouží k odhadu nejpravděpodobnějšího stavu systému kombinací výstupů více senzorů. Výstupy senzorů jsou zatíženy chybami. „*Kalmanův filtr se tedy používá na přesné určení stavových proměnných, t.j. na optimální potlačení měřeného šumu*“ [8, s. 149]. Je-li dynamický systém popsán nelineárními diferenciálními rovnicemi, používají se různě modifikované varianty Kalmanova filtru, jako linearizovaný nebo rozšířený Kalmanův filtr, viz [8] nebo [4].

2.4.1 Rozšířený Kalmanův filtr (EKF)

Nelineární diskretní dynamický systém je popsán rovnicí

$$\mathbf{x}_k = \mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \quad (2.1)$$

kde \mathbf{x}_k značí stav systému, \mathbf{u}_k je vektor řídicích vstupů a \mathbf{w}_k značí šum s nulovou střední hodnotou a s kovarianční maticí \mathbf{Q}_k . Na diagonále kovarianční matice jsou rozptyly šumů jednotlivých stavových proměnných a mimo diagonálu jsou kovariance každých dvou. Matice \mathbf{Q}_k vlastně vyjadřuje jak moc důvěřujeme modelu systému. Kdyby byla nulová, znamenalo by to, že modelu věříme absolutně, tedy že do stavového vektoru zahrnujeme všechny náhodné veličiny, které mohou systém ovlivňovat, že jsme správně popsali jejich závislosti a vývoj v čase.

Rovnice

$$\mathbf{y}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k \quad (2.2)$$

značí nelineární vztah měření \mathbf{y}_k se stavem systému \mathbf{x}_k . Vektor \mathbf{v}_k je šum s nulovou střední hodnotou a s kovarianční maticí \mathbf{R}_k , která vlastně vyjadřuje míru důvěry v přesnost měření.

Kalmanův filtr si uchovává poslední stav systému \mathbf{x}_k a nejistotu odhadu tohoto stavu vyjádřenou kovarianční maticí \mathbf{P}_k . Po inicializaci \mathbf{x}_0 a \mathbf{P}_0 se opakovaně provádí dva kroky:

1. **Predikce**, při které se spočítá další odhad stavu systému jako $\mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_k)$ a aktualizuje se kovarianční matice \mathbf{P}_k^- (horní index $-$ značí odhad po predikci, odhad po korekci je označen horním indexem $+$, stříška nad stavovým vektorem znamená odhad).
2. **Korekce**, při které se provedené měření kombinuje s odhadem po predikci.

Dále jsou sepsány rovnice, které jsou počítané rozšířeným Kalmanovým filtrem v těchto dvou krocích. Detailně je algoritmus EKF popsán v [8] a [4].

Predikce Během predikce se provádí linearizace funkce \mathbf{f}_{k-1} .

$$\mathbf{F}_{k-1} = \left. \frac{\partial \mathbf{f}_{k-1}(\mathbf{x}, \mathbf{u}_k)}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^+} \quad (2.3)$$

Dále je odhadnut stav podle modelu dynamického systému.

$$\hat{\mathbf{x}}_k^- = \mathbf{f}_{k-1}(\hat{\mathbf{x}}_{k-1}^+, \mathbf{u}_k) \quad (2.4)$$

Nakonec se aktualizuje kovarianční matice \mathbf{P}_k .

$$\mathbf{P}_k^- = \mathbf{F}_{k-1} \mathbf{P}_{k-1}^+ \mathbf{F}_{k-1}^T + \mathbf{Q}_{k-1} \quad (2.5)$$

Z rovnice 2.5 je vidět, že bez korekce by se nejistota odhadu stále zvětšovala.

Korekce Během korekce je třeba provést linearizaci funkce \mathbf{h}_k .

$$\mathbf{H}_k = \left. \frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_k^-} \quad (2.6)$$

Dále se provede výpočet tzv. Kalmanova zesílení.

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.7)$$

Kalmanovo zesílení \mathbf{K}_k je důležité pro následnou samotnou korekci predikovaného stavu systému a určuje jak moc se má odhad korigovat změřenými hodnotami. Jeho hodnota roste s tím, jak roste důvěra v měření (vyjádřena v kovarianční matici \mathbf{R}_k) nebo klesá důvěra ve správnou predikci (vyjádřena v \mathbf{P}_k^-).

Dalším krokem je korekce stavového vektoru

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)) , \quad (2.8)$$

kde $\mathbf{y}_k - \mathbf{h}_k(\hat{\mathbf{x}}_k^-)$ je rozdíl mezi provedeným měřením a jeho odhadem. Tento rozdíl je zesílen vzhledem k důvěře v měření/predikci a přičten k predikovanému stavu.

Nakonec je provedena aktualizace kovarianční matice \mathbf{P}_k .

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (2.9)$$

Při použití rovnice (2.9) je dále nutné zkontrolovat symetrii a pozitivní definitnost matice \mathbf{P}_k^+ . Symetrie lze zajistit například následným výpočtem

$$\mathbf{P}_k^+ = \frac{1}{2} (\mathbf{P}_k^+ + \mathbf{P}_k^{+T}) \quad (2.10)$$

2.4.2 Josephův tvar korekce kovarianční matice

Rovnice

$$\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (2.11)$$

je tzv. *Josephův tvar* korekce kovarianční matice. Zatímco po použití rovnice (2.9) je dále potřeba provést kontrolu matice \mathbf{P}_k^+ na symetrii a pozitivní definitnost, po použití rovnice (2.11) to nutné není, protože v ní jsou sčítány dvě symetrické matice, z nichž jedna je pozitivně definitní a druhá pozitivně semidefinitní. Výsledná kovarianční matice \mathbf{P}_k^+ je tedy symetrická a pozitivně definitní (viz [3, kap. 4]).

Rovnice (2.11) je výpočetně náročnější než (2.9). Níže v této kapitole jsou popsány metody pro snížení výpočetní náročnosti při použití Josephova tvaru.

2.4.3 Výpočet Kalmanova zesílení pomocí UD rozkladu

Nejvíce výpočetně náročná část standardního Kalmanova filtru je inverze v rovnici (2.7) pro určení Kalmanova zesílení. Podle [3] je nejvhodnější se v běžném EKF inverzi vyhnout pomocí UD rozkladu.

UD rozklad Je-li matice $\mathbf{M} \in \mathbb{R}^{n \times n}$ symetrická, pozitivně definitní, pak ji lze rozložit pomocí UD rozkladu (též známý jako modifikovaný Choleského rozklad) jako

$$\mathbf{M} = \mathbf{U} \mathbf{D} \mathbf{U}^T,$$

kde \mathbf{U} je matice v horním trojúhelníkovém tvaru s jedničkami na diagonále a \mathbf{D} je diagonální matice. Matice \mathbf{U} a \mathbf{D} se získají následovně:

Pro $j = n, \dots, 1$ a $i = j, \dots, 1$:

$$u_{j,i} = \begin{cases} 1 & \text{když } j = i \\ \frac{1}{d_{j,j}} \left(m_{i,j} - \sum_{k=j+1}^n u_{i,k} \cdot d_{k,k} \cdot u_{j,k} \right) & \text{když } j \neq i \end{cases} \quad (2.12)$$

$$d_{j,i} = \begin{cases} m_{i,j} - \sum_{k=j+1}^n u_{i,k} \cdot d_{k,k} \cdot u_{j,k} & \text{když } j = i \\ 0 & \text{když } j \neq i \end{cases}$$

kde n je počet řádků a sloupců matice \mathbf{M} .

V [3] je doporučeno rozložit výraz $\mathbf{H} \mathbf{P} \mathbf{H}^T + \mathbf{R}$ na $\mathbf{U} \mathbf{D} \mathbf{U}^T$ UD dekompozicí a potom vyřešit rovnici

$$\mathbf{U} \mathbf{D} \mathbf{U}^T \mathbf{X} = \mathbf{H}, \quad (2.13)$$

kde neznámé $\mathbf{X} = (\mathbf{H}\mathbf{P}\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}$. Matice \mathbf{X} lze nalézt postupným vyřešením rovnic

$$\begin{aligned}\mathbf{U}\mathbf{X}_1 &= \mathbf{H} \\ \mathbf{D}\mathbf{X}_2 &= \mathbf{X}_1 \\ \mathbf{U}^T\mathbf{X} &= \mathbf{X}_2\end{aligned}\tag{2.14}$$

Matice \mathbf{U} je v trojúhelníkovém tvaru, proto lze první a poslední rovnici vyřešit postupným dosazováním. Jelikož \mathbf{D} je diagonální, druhá maticová rovnice je systémem nezávislých skalárních rovnic, kde i -tý řádek \mathbf{X}_2 se rovná i -tému řádku \mathbf{X}_1 vydělenému číslem $d_{i,i}$. Pro podrobnosti viz [3, kap. 6].

2.4.4 Sekvenční korekce

Dalším způsobem, jak se vyhnout výpočtu inverze v rovnici (2.7) je sekvenční korekce. Následující popis této metody čerpá z [3, kap. 6] a z [2].

Je-li kovarianční matice měření \mathbf{R} diagonální, je možné metodu použít bez dekorelace šumu měření (viz níže). Kroky predikce, tedy rovnice (2.4) až (2.5), a první krok korekce, tedy rovnice (2.6), zůstávají nezměněny. Rovnice (2.7) až (2.11) běžného EKF jsou nahrazeny iterativním výpočtem:

$$\text{Inicializace: } \hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^-, \mathbf{P}_k^0 = \mathbf{P}_k^-$$

Pro $i = 1, \dots, m$:

$$\begin{aligned}\mathbf{k}_k^i &= \frac{\mathbf{P}_k^{i-1}\mathbf{h}_k^i}{\mathbf{h}_k^{iT}\mathbf{P}_k^{i-1}\mathbf{h}_k^i + r_{i,i}} \\ \hat{\mathbf{x}}_k^i &= \hat{\mathbf{x}}_k^{i-1} + \mathbf{k}_k^i \left(y_k^i - \mathbf{h}_k^{iT}\hat{\mathbf{x}}_k^{i-1} \right) \\ \mathbf{P}_k^i &= \left(\mathbf{I} - \mathbf{k}_k^i\mathbf{h}_k^{iT} \right) \mathbf{P}_k^{i-1}\end{aligned}\tag{2.15}$$

kde m je počet řádků matice \mathbf{H}_k , \mathbf{h}_k^{iT} je její i -tý řádek, y_k^i je i -tá položka vektoru měření a $r_{i,i}$ je prvek matice \mathbf{R} na i -tém řádku a i -tém sloupci. Potom $\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^m$ a $\mathbf{P}_k^+ = \mathbf{P}_k^m$.

Jestliže \mathbf{R} není diagonální, je třeba nejprve provést dekorelaci šumu měření.

Dekorelace šumu měření Nejdříve se provede UD rozklad matice \mathbf{R} a poté se naleznou transformace vektoru měření a matice \mathbf{H} vyřešením rovnic

$$\begin{aligned}\mathbf{U}\tilde{\mathbf{y}}_k &= \mathbf{y}_k \\ \mathbf{U}\tilde{\mathbf{H}}_k &= \mathbf{H}_k\end{aligned}$$

pro neznámé $\tilde{\mathbf{y}}_k$ a $\tilde{\mathbf{H}}_k$ postupným dosazováním (matice \mathbf{U} je v horním trojúhelníkovém tvaru s jedničkami na diagonále). Potom lze provést sekvenční korekci podle rovnic (2.15), pouze s tím rozdílem, že se prvky vektoru měření \mathbf{y}_k nahradí prvky transformovaného vektoru měření $\tilde{\mathbf{y}}_k$, řádky matice \mathbf{H}_k se nahradí řádky $\tilde{\mathbf{H}}_k$ a místo kovarianční matice \mathbf{R} se bude počítat s maticí \mathbf{D} z UD rozkladu.

Tato metoda redukuje potřebu počítání inverze, což je výpočetně náročné na jednoduché dělení skalárem. Pro podrobnosti o de Korelaci šumu měření viz [3, kap. 6].

2.4.5 Bierman-Joseph

Korekce kovarianční matice \mathbf{P}_k^+ podle Josephova tvaru (rovnice (2.11)) je výpočetně náročnější než standardní korekce podle rovnice (2.9). Jedna z metod jak tento výpočet urychlit je Biermanova metoda. Tato metoda provádí korekci sekvenčně po jednotlivých měření. Během výpočtu se uchovávají některé mezivýpočty pro následné znovupoužití. Dále je popsán algoritmus výpočtu:

$$\begin{aligned}
& \text{Inicializace: } \hat{\mathbf{x}}_k^0 = \hat{\mathbf{x}}_k^-, \mathbf{P}_k^0 = \mathbf{P}_k^- \\
& \text{Pro } i = 1, \dots, m : \\
& \quad \mathcal{T}_1 = \sqrt{r_{i,i}} \\
& \quad \mathcal{T}_2 = \frac{\mathbf{h}_k^{i,T}}{\mathcal{T}_1} \\
& \quad \mathcal{T}_3 = \frac{y_k^i}{\mathcal{T}_1} - \mathcal{T}_2 \hat{\mathbf{x}}_k^{i-1} \\
& \quad \mathcal{T}_4 = \mathbf{P}_k^{i-1} \mathcal{T}_2^T \\
& \quad \mathcal{T}_5 = \mathcal{T}_2 \mathcal{T}_4 + 1 \\
& \quad \mathcal{T}_6 = \frac{\mathcal{T}_4}{\mathcal{T}_5} = \mathbf{k}_k^i \\
& \quad \mathcal{T}_7 = \mathbf{P}_k^{i-1} - \mathcal{T}_6 \mathcal{T}_4^T \\
& \quad \mathcal{T}_8 = \mathcal{T}_7 \mathcal{T}_2^T \\
& \quad \mathbf{P}_k^i = \mathcal{T}_7 - \mathcal{T}_8 \mathcal{T}_6 + \mathcal{T}_6 \mathcal{T}_6^T \\
& \quad \hat{\mathbf{x}}_k^i = \hat{\mathbf{x}}_k^{i-1} + \mathcal{T}_6 \mathcal{T}_3 \\
& \text{Potom } \hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^m \text{ a } \mathbf{P}_k^+ = \mathbf{P}_k^m
\end{aligned} \tag{2.16}$$

kde m je počet řádků matice \mathbf{H}_k , $\mathbf{h}_k^{i,T}$ je její i -tý řádek a y_k^i je i -tá položka vektoru měření. Popis algoritmu je převzat z [3, kap. 6] po úpravě rovnice pro výpočet \mathcal{T}_3 . V [3] není y_k^i vydělené \mathcal{T}_1 , což je chyba způsobující špatný výpočet $\hat{\mathbf{x}}_k^+$.

Je-li zobrazení stavového vektoru do vektoru měření nelineární, je potřeba rovnici pro \mathcal{T}_3 upravit na tvar

$$\mathcal{T}_3 = \frac{y_k^i - \mathfrak{h}_k(\hat{\mathbf{x}}_k^{i-1}, i)}{\mathcal{T}_1}, \tag{2.17}$$

kde \mathfrak{h}_k je nelineární funkce a $\mathfrak{h}_k(\hat{\mathbf{x}}_k^{i-1}, i)$ je i -tý prvek předpokládaného vektoru měření odhadnutého na základě hodnoty predikovaného stavového vektoru.

Biermanova metoda, tak jak je popsána výše, vyžaduje de Korelovaný šum měření, tedy diagonální kovarianční matici \mathbf{R} . Není-li \mathbf{R} diagonální, je třeba nejdříve provést de Korelaci šumu měření tak, jak je popsána v 2.4.4.

2.4.6 De Vries-Joseph

V [3, kap. 6] je kromě Biermanovy metody popsána ještě další metoda pro výpočet korekčního kroku Kalmanova filtru s použitím Josephova tvaru (2.11) aktualizace matice \mathbf{P}_k^-

- De Vriesova metoda. Stejně jako Biermanova si uchovává mezivýsledky, které dále znovu používá. Níže jsou popsány dvě varianty De Vriesovy metody podle [3]. První varianta je s použitím UD rozkladu. Výpočet probíhá následovně:

$$\begin{aligned}
\mathcal{F}_1 &= \mathbf{P}_k^- \mathbf{H}^T \\
\mathcal{F}_2 &= \mathbf{H} \mathcal{F}_1 + \mathbf{R} \\
\mathbf{U} \mathbf{D} \mathbf{U}^T &= \mathcal{F}_2 \\
\mathbf{U} \mathbf{D} \mathbf{U}^T \mathbf{K}_k^T &= \mathcal{F}_1^T \quad (\text{řešení soustavy rovnic pro neznámé } \mathbf{K}_k^T) \\
\mathcal{F}_3 &= \frac{1}{2} \mathbf{K}_k \mathcal{F}_2 - \mathcal{F}_1 \\
\mathcal{F}_4 &= \mathcal{F}_3 \mathbf{K}_k^T \\
\mathbf{P}_k^+ &= \mathbf{P}_k^- + \mathcal{F}_4 + \mathcal{F}_4^T \\
\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H} \hat{\mathbf{x}}_k^- \right)
\end{aligned} \tag{2.18}$$

Výpočet neznámého \mathbf{K}_k^T ve čtvrté rovnici lze provést postupným vyřešením tři soustav rovnic (2.14).

Druhá varianta využívá sekvenční korekce. Jako ostatní sekvenční postupy předpokládá nekorelovaný šum měření. Je-li šum korelovaný, je třeba nejdříve provést jeho dekorelaci (popsáno v 2.4.4). Kroky výpočtu sekvenční varianty De Vriesovy metody jsou:

$$\begin{aligned}
\text{Inicializace: } \hat{\mathbf{x}}_k^0 &= \hat{\mathbf{x}}_k^-, \quad \mathbf{P}_k^0 = \mathbf{P}_k^- \\
\text{Pro } i &= 1, \dots, m : \\
\mathcal{F}_1 &= \mathbf{P}_k^{i-1} \mathbf{h}_k^i \\
\mathcal{F}_2 &= \mathbf{h}_k^{iT} \mathcal{F}_1 + r_{i,i} \\
\mathbf{k}_k^i &= \frac{\mathcal{F}_1}{\mathcal{F}_2} \\
\mathcal{F}_3 &= \frac{1}{2} \mathbf{k}_k^i \mathcal{F}_2 - \mathcal{F}_1 \\
\mathcal{F}_4 &= \mathcal{F}_3 \mathbf{k}_k^{iT} \\
\mathbf{P}_k^i &= \mathbf{P}_k^{i-1} + \mathcal{F}_4 + \mathcal{F}_4^T \\
\hat{\mathbf{x}}_k^i &= \hat{\mathbf{x}}_k^{i-1} + \mathbf{k}_k \left(y_k^i - \mathbf{h}_k^{iT} \hat{\mathbf{x}}_k^{i-1} \right) \\
\text{Potom } \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^m \quad \text{a} \quad \mathbf{P}_k^+ = \mathbf{P}_k^m
\end{aligned} \tag{2.19}$$

kde m je počet řádků matice \mathbf{H}_k , \mathbf{h}_k^{iT} je její i -tý řádek a y_k^i je i -tá položka vektoru měření.

3 Navigační jednotka

V této kapitole je popsán INS/GPS navigační systém, jehož software je v rámci bakalářské práce optimalizován.

3.1 Hardware

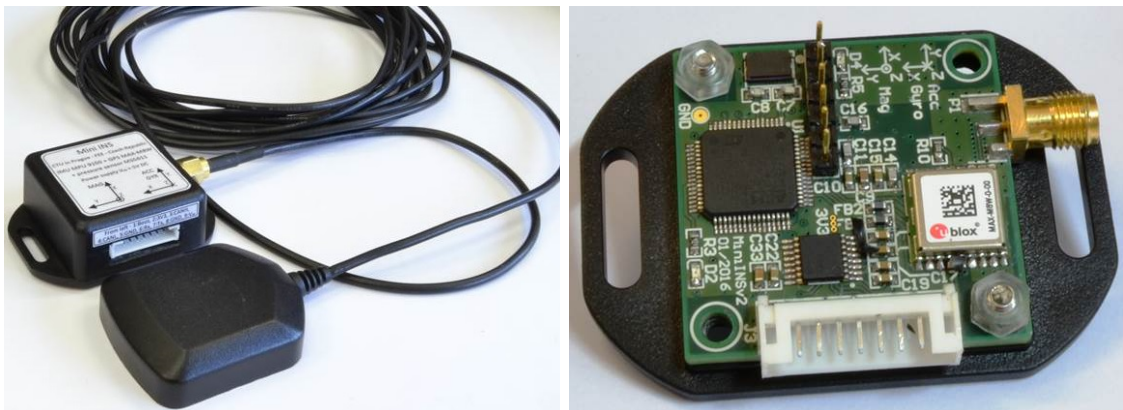
Navigační systém se skládá z inerciální měřicí jednotky InvenSense MPU-9150¹, GNSS modulu u-blox MAX-M8² a mikrokontroléru STM32F405³. Fotografie navigační jednotky jsou na obrázku 3.1.

IMU

InvenSense MPU-9150 (viz [5]) obsahuje tříosý kapacitní MEMS akcelerometr (viz 2.2.3) s rozsahy od $\pm 2g$ po $\pm 16g$, tříosý vibrační MEMS senzor úhlové rychlosti (viz 2.2.4) s rozsahy ± 250 až ± 2000 $^{\circ}/s$ a tříosý magnetometr s rozsahem ± 1200 μT . Výstupy akcelerometrů a senzorů úhlové rychlosti jsou digitalizovány šesti 16 bitovými A/D převodníky, výstupy magnetometru pak třemi 13 bitovými.

GNSS modul

u-blox MAX-M8 (viz [10]) je GNSS modul schopný přijímat signály GPS, GLONASS a BeiDou. Pro vyšší výkonost je možné přijímat a kombinovat dva signály zároveň. Přesnost v určení polohy je uváděna jako 2 metry CEP (tzn. v 50% případech se změřená poloha liší od skutečné o maximálně 2 metry). Obnovovací frekvence je až 18 Hz při přijímání signálu z jednoho GNSS a až 10 Hz při současném přijímání dvou signálů.



Obrázek 3.1: Navigační jednotka

¹<http://www.invensense.com/products/motion-tracking/9-axis/mpu-9150/>

²<https://www.u-blox.com/en/product/max-m8-series>

³<http://www.st.com/web/en/catalog/mmc/FM141/SC1169/SS1577/LN1035>

Mikrokontrolér

STM32F405 je mikrokontrolér založený na 32-bitovém jádře ARM Cortex-M4, které může pracovat na frekvenci až 168 MHz. Obsahuje matematický koprocessor (*Floating Point Unit* - FPU), který urychluje výpočty čísel s plovoucí řádovou čárkou v jednoduché přesnosti (*single precision*).

Dále mikrokontrolér podporuje jednocyklové DSP instrukce pro MAC (*Multiply and accumulate*) a SIMD (*Single instruction, multiple data*) operace. MAC je operace typu $a \leftarrow a + b \times c$, kdy se spočítaný výsledek násobení přičte do akumulátoru v jednom cyklu. Během jedné SIMD instrukce je možné provést jednu operaci nad více daty. Tento mikrokontrolér slouží v použité navigační jednotce pro výpočet Kalmanova filtru.

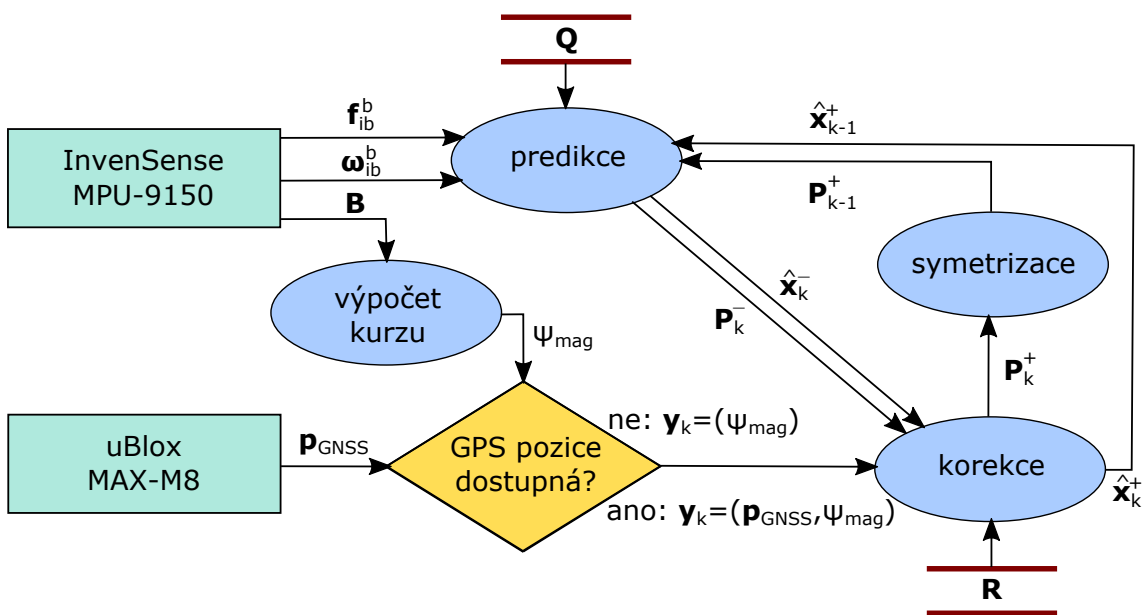
3.2 Neoptimalizovaný výpočet EKF

V neoptimalizovaném programu navigačního systému je počítán standardní EKF tak, jak byl popsán výše rovnicemi (2.4) až (2.9). Po každém korekčním kroku je navíc provedena symetrizace \mathbf{P}_k podle rovnice (2.10). Inverzní matice v rovnici (2.7) je počítána pomocí LU dekompozice. Diagram datového toku neoptimalizovaného programu je znázorněn na obrázku 3.2.

Reálná čísla v maticích a vektorech jsou reprezentovány datovým typem *float*, tedy jako čísla s pohyblivou řádovou čárkou v jednoduché přesnosti (32 bitů). To znamená, že může být při výpočtu využito matematického koprocessoru, který je součástí mikroprocesoru z navigační jednotky (viz výše).

Program počítá EKF pro stavový prostor dimenze 12. Stavový vektor vypadá následovně:

$$\mathbf{x} = [\mathbf{p}, \mathbf{v}, \Psi, \mathbf{b}]^T,$$



Obrázek 3.2: Diagram datového toku

kde \mathbf{p} je vektor pozice, \mathbf{v} vektor rychlosti, \mathbf{b} vektor biasů senzoru úhlové rychlosti a Ψ je vektor Eulerových úhlů $[\phi, \theta, \psi]$. Korekční krok se provádí po každé predikci. Jsou-li k dispozici data z GPS, vektor měření má čtyři prvky:

$$\mathbf{y} = [\mathbf{p}_{GPS}, \psi_{mag}]^T,$$

kde \mathbf{p}_{GPS} je vektor pozice ve 3D prostoru získaný z GPS a ψ_{mag} je kurz vypočítaný na základě údajů z magnetometru. Nejsou-li k dispozici data z GPS, vektor měření má pouze jediný prvek, a to ψ_{mag} .

Při běhu neoptimalizovaného programu na procesoru Cortex-M4 nad daty ze souboru 20141031_Flight_01_DATA.mat (viz CD) trvá jeden cyklus rozšířeného Kalmanova filtru průměrně 9631 μs . Poměr korekcí s pozicí z GPS ku korekcím bez pozice z GPS je při použití dat z tohoto souboru 1:19.

4 Optimalizace

V této kapitole jsou popsány návrhy optimalizace výpočtu EKF. V následující kapitole je pak porovnána efektivita těchto návrhů.

4.1 ARM CMSIS DSP Library

CMSIS (*Cortex Microcontroller Software Interface Standard*) je sada API a knihoven pro procesory ARM Cortex-M. Jedna ze součástí CMSIS je i DSP knihovna¹. DSP knihovna obsahuje přes 60 různých funkcí rozdělených do několika kategorií (transformace, maticové operace, filtry, statistické funkce atd.). Verze pro Cortex-M4 a Cortex-M7 jsou optimalizované pro SIMD instrukce a využití FPU, kterými disponují na rozdíl od jiných Cortex-M procesorů.

Jelikož se v EKF počítá s maticemi, tak jedním z cílů optimalizace je porovnání efektivity výpočtu EKF pomocí maticových operací z DSP knihovny s efektivitou výpočtu EKF původním neoptimalizovaným programem.

Většina zdrojového kódu v jazyce C, který vznikl jako součást této bakalářské práce, je psána tak, aby se pouhou definicí makra `ARM_DSP_LIBRARY` šlo přepnout do režimu používání CMSIS DSP knihovny a jeho smazáním zpět do režimu používání původní implementace maticových operací ze souboru `matrix_operation.c`. Rovněž původní kód Kalmanova filtru v souboru `kalman_my.c` byl přepsán tak, aby umožnil toto přepínání.

Základem pro možnost přepínání je podmíněná definice maker v `library_switch.h` následujícím způsobem:

```
#ifndef ARM_DSP_LIBRARY
#define OPERACE(P)      funkce_1(P);
#else
#define OPERACE(P)      funkce_2(P);
#endif
```

Potom při překladau preprocesor nahradí makro `OPERACE` funkcí `funkce_1` z DSP knihovny nebo funkcí `funkce_2` ze souboru `matrix_operation.c` v závislosti na tom, zda je, či není definováno makro `ARM_DSP_LIBRARY`.

4.2 Korekční krok EKF

Výpočetně nejnáročnější část klasického EKF, který je implementován v neoptimalizovaném programu navigačního systému, je počítání inverzní matice v rovnici (2.7). V rozboru je popsáno několik dalších možností, jak provést korekční krok Kalmanova filtru. V rámci bakalářské práce byly implementovány tyto metody korekce:

¹<http://www.keil.com/pack/doc/CMSIS/DSP/html/index.html>

1. inverze pomocí UD dekompozice (2.4.3),
2. sekvenční korekce (2.4.4),
3. Bierman-Joseph (2.4.5),
4. De Vries-Joseph s UD dekompozicí (2.4.6),
5. De Vries-Joseph, sekvenční. (2.4.6).

Porovnání jejich výpočetních náročností pro různé dimenze prostoru stavů a měření je v kapitole „Výsledky“.

4.2.1 Implementace

Implementace těchto metod je na přiloženém CD v souboru `correction.c`. Přepínat mezi jednotlivými metodami lze v programu změnou hodnoty makra `CORRECTION_TYPE`, které je definované v souboru `main.h`. V tabulce 4.1 jsou vypsány názvy funkcí realizující danou metodu korekce a k nim příslušící hodnota makra `CORRECTION_TYPE`. Funkce `ekf_measupd_multiaid` z původního programu není na rozdíl od ostatních definována v souboru `correction.c` ale v `kalman_my.c`.

Metoda	Funkce	CORRECTION_TYPE
inverze pomocí UD dekompozice	<code>ekf_ud_correction</code>	1
De Vries-Joseph s UD rozkladem	<code>ekf_ud_devries_correction</code>	2
sekvenční korekce	<code>ekf_seq_correction</code>	3
De Vries-Joseph, sekvenční	<code>ekf_seq_devries_correction</code>	4
Bierman-Joseph	<code>ekf_seq_bierman_correction</code>	5
původní s LU rozkladem	<code>ekf_measupd_multiaid</code>	jinak

Tabulka 4.1: Funkce realizující korekční metody

Všechny nově vzniklé funkce pro korekci mají prototyp shodný (mimo jména) s prototypem:

```
int func(MATRIX *H, MATRIX *R, MATRIX *y, MATRIX *P, MATRIX *x);
```

V případě, že funkce dobehnou bez chyb, vracejí 0, když se během výpočtu chyba naskytne, vracejí 1. Funkce přijímají jako parametry 5 ukazatelů na matice. `MATRIX` je makro podmínečně definované v `library_switch.h`. Při použití původní implementace maticových operací jsou matice reprezentované strukturou `matrix` definovanou v `matrix_operation.h`. Je-li definováno makro `ARM_DSP_LIBRARY` je použita DSP knihovna a její struktura s názvem `arm_matrix_instance_f32`. Význam argumentů:

- **H**: Jakobián funkce **h**,
- **R**: kovarianční matice šumu měření,
- **y**: vektor měření,
- **P**: kovarianční matice modelu systému \mathbf{P}_k^- ,
- **x**: predikce stavového vektoru $\hat{\mathbf{x}}_k^-$.

Funkce různým způsobem spočítají Kalmanovo zesílení \mathbf{K}_k a na místě korigují matici \mathbf{P}_k^- a stavový vektor $\hat{\mathbf{x}}_k^-$ - tzn. po doběhnutí funkce ukazují ukazatelé \mathbf{P} a \mathbf{x} (předané jako argumenty) na \mathbf{P}_k^+ a $\hat{\mathbf{x}}_k^+$.

4.3 Predikce \mathbf{P}_k pomocí operací se submaticemi

Matice \mathbf{F}_k (Jakobián funkce \mathbf{f}_k z popisu dynamického systému (2.1)) může obsahovat velké množství nulových prvků. To je dáno tím, že časový vývoj stavových veličin není zpravidla závislý na všech ostatních stavových veličinách, ale pouze na některých z nich. Při počítání rovnice (2.5) tak dochází k velkému počtu násobení a přičítání nul, nijak neovlivňující výsledek. To vede k myšlence rozdělit matici \mathbf{F}_k na submatice, z nichž některé jsou nulové nebo jednotkové a implementovat maticové násobení, kde by se nejdříve operovalo se submaticemi a ne hned s jednotlivými prvky, přičemž kdyby byla jedna ze submatic nulová nebo jednotková, dalo by se vyhnout volání funkce pro součin.

Matice \mathbf{F}_k v původním programu navigační jednotky je prvkem $\mathbb{R}^{12 \times 12}$. Konkrétně tuto matici lze reprezentovat jako matici matic z $\mathbb{R}^{6 \times 6}$ nebo z $\mathbb{R}^{3 \times 3}$ následovně:

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_k^{1,1} & \mathbf{F}_k^{1,2} \\ \mathbf{0}_{6 \times 6} & \mathbf{F}_k^{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{F}_k^{1,2} & \mathbf{F}_k^{1,3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{F}_k^{2,2} & \mathbf{F}_k^{2,3} & \mathbf{F}_k^{2,4} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{F}_k^{3,3} & \mathbf{F}_k^{3,4} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}, \quad (4.1)$$

kde \mathbf{I} je jednotková a $\mathbf{0}$ je nulová matice. Stejně tak lze reprezentovat i matice \mathbf{P}_k .

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_k^{1,1} & \mathbf{P}_k^{1,2} \\ \mathbf{P}_k^{2,1} & \mathbf{P}_k^{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{P}_k^{1,1} & \mathbf{P}_k^{1,2} & \mathbf{P}_k^{1,3} & \mathbf{P}_k^{1,4} \\ \mathbf{P}_k^{2,1} & \mathbf{P}_k^{2,2} & \mathbf{P}_k^{2,3} & \mathbf{P}_k^{2,4} \\ \mathbf{P}_k^{3,1} & \mathbf{P}_k^{3,2} & \mathbf{P}_k^{3,3} & \mathbf{P}_k^{3,4} \\ \mathbf{P}_k^{4,1} & \mathbf{P}_k^{4,2} & \mathbf{P}_k^{4,3} & \mathbf{P}_k^{4,4} \end{bmatrix}$$

Prvek (1,1) součinu $\mathbf{M} = \mathbf{F}\mathbf{P}$ se submaticemi z $\mathbb{R}^{3 \times 3}$ je tedy

$$\mathbf{M}^{1,1} = \mathbf{I}_{3 \times 3} \mathbf{P}_k^{1,1} + \mathbf{F}_k^{1,2} \mathbf{P}_k^{2,1} + \mathbf{F}_k^{1,3} \mathbf{P}_k^{3,1} + \mathbf{0}_{3 \times 3} \mathbf{P}_k^{4,1} \quad (4.2)$$

Ze čtyř maticových násobení v (4.2) se lze (konkrétně v tomto případě) dvěma vyhnout, protože $\mathbf{I}_{3 \times 3} \mathbf{P}_k^{1,1} = \mathbf{P}_k^{1,1}$ a $\mathbf{0}_{3 \times 3} \mathbf{P}_k^{4,1} = \mathbf{0}_{3 \times 3}$. Implementované násobení se submaticemi se vyhýbá volání maticového násobení, když je jedna ze submatic jednotková nebo nulová, což může vést ke snížení výpočetního času.

4.3.1 Implementace

Funkce pro operace s maticemi matic potřebné pro výpočet rovnice (2.5) jsou implementované v souboru `hypermatrix.c`. Při použití DSP knihovny je struktura matice matic v `hypermatrix.h` definována jako:

```
typedef struct {
    unsigned size;
    unsigned subsizes;
    arm_matrix_instance_f32 **submat;
} hypermatrix;
```

kde `size` je počet submatic v řádku/sloupci, `subsize` je velikost každé submatice a `submat` je pole ukazatelů na submatice. Nulové a jednotkové submatice jsou reprezentované ukazateli do, pro tento účel, alokované paměti. Převod ARM matice do matice matic vyžaduje zásah do paměti, protože klasická ARM matice je uložena po řádcích, kdežto matici matic je nutné (pro správnou funkci funkcí z DSP knihovny volaných na submatice) do paměti uložit po jednotlivých submaticích.

Není-li použita DSP knihovna, struktura `hypermatrix` vypadá následovně:

```
typedef struct {
    unsigned size;
    unsigned subsize;
    unsigned info [SIZE] [SIZE];
    matrix submat;
} hypermatrix;
```

Význam `size` a `subsize` je stejný jako v předchozím případě. Pole `info` má tolik prvků kolik má matice submatic. Je-li prvek (i, j) pole `info` rovno 1, submatice v i -tém řádku a j -tém sloupci je jednotková, je-li tento prvek roven 2, submatice je nulová.

V tomto případě je převod mezi maticí a maticí matic jednoduchý. Celá matice ve struktuře `matrix` se přidá do struktury matice matic `hypermatrix`. K přístupu k jednotlivým submaticím se pak využijí proměnné `from` a `to`, které jsou součástí struktury `matrix` a které uchovávají informaci o tom, odkud kam v alokovaném dvourozměrném poli pro matici jsou skutečně uloženy prvky matice (index řádku a sloupce levého horního rohu matice a index řádku a sloupce pravého dolního rohu). V implementovaných operacích se submaticemi tak stačí v tomto případě pouze měnit hodnoty `to` a `from` a tím tak „ukazovat“ na submatice přímo ve struktuře matice.

V rámci optimalizace byla implementována možnost spočítat rovnici (2.5) pomocí operací se submaticemi. Funkce realizující výpočet se jmenuje `hypmat_predictP`.

4.4 Předpočítávání

Při optimalizaci programu byla snaha o předpočítání a uchování mezivýsledků, které jsou dále vícekrát použity. Největší úsporu výpočetního času pomocí předpočítávání (viz „Výsledky“) přineslo předpočítání goniometrických funkcí Eulerových úhlů ve funkcích realizujících aktualizaci matice \mathbf{F} , přechod dynamického systému do dalšího stavu nebo přepočítání transformační matice mezi souřadnicovým systémem navigovaného tělesa a NED.

5 Výsledky

V této kapitole je porovnávána efektivita různých metod pro optimalizaci výpočtu EKF. Tabulky s přesnými naměřenými hodnotami a grafy jsou v příloze B. Grafy byly vytvořeny Matlabem. Skripty pro tvorbu grafů jsou na přiloženém CD.

5.1 Způsob měření

Všechny měřené výpočty byly prováděny na STM32 F4 Discovery Kitu, který obsahuje mikrokontrolér STM32F407, jenž je založený na jádře ARM Cortex-M4 stejně jako mikrokontrolér STM32F405, který je součástí optimalizovaného navigačního systému.

Měření času běhu jednotlivých funkcí bylo prováděno za pomoci 32-bitového timeru 5. V [9, str. 217] se uvádí, že hodiny timeru běží

- a) na stejné frekvenci jako hodiny APB domény, ke které je timer připojen, je-li APB prescaler roven 1.
- b) na dvojnásobné frekvenci hodin APB domény, ke které je timer připojen, není-li APB prescaler roven 1.

Timer 5 je součástí APB1 domény. APB1 prescaler je nastaven na 4 a systémové hodiny jsou nakonfigurované na frekvenci 168 MHz, frekvence hodin APB1 domény je tedy 42 MHz a frekvence hodin timeru f_{tim} je podle bodu b) 84 MHz.

Frekvence čítání f_{cnt} lze nastavit uložením hodnoty do registru TIM5_PSC. Pak podle [9]

$$f_{cnt} = \frac{f_{tim}}{TIM5_PSC + 1}$$

Tedy uložením hodnoty 83 je dosaženo frekvence čítání 1 MHz. Právě na tuto frekvenci byl při měření timer 5 nastaven. Měření tedy probíhalo s rozlišením 1 mikrosekundy.

Inicializace timeru a funkce pro jeho spuštění a zastavení jsou implementované v souboru `timer5.c`.

5.2 Korekční krok EKF

V této části jsou porovnány časové efektivity korekčních metod zmíněných v kapitole 4.2. V následujícím textu je stavový prostor označen jako \mathcal{S} a prostor měření jako \mathcal{M} pro jednodušší popis počtu prvků vektorů stavů a měření jako $\dim \mathcal{S}$ a $\dim \mathcal{M}$. Testovací program korekčních metod je v souboru `correction_test.c`.

Na následujících grafech je na ose y počet prvků vektoru měření a na ose x je počet prvků stavového vektoru. Na obrázcích jsou vidět vrstevnice času. Na některých vrstevnicích je uveden čas v mikrosekundách. Grafy byly vytvořeny v Matlabu pomocí funkce `contour`, viz soubor `graph.m` na přiloženém CD.

5.2.1 Měření

Měřena byla doba výpočtu korekčního kroku KF bez doby linearizace funkce \mathbf{h}_k (2.6). Tento výpočet se mezi korekčními metodami nijak neliší, trvá u všech metod stejně dlouho a neovlivňuje tak porovnání časových náročností metod.

Do měření je tedy u všech metod započítána doba výpočtu Kalmanova zesílení \mathbf{K}_k , aktualizace stavového vektoru $\hat{\mathbf{x}}_k$ a aktualizace kovarianční matice \mathbf{P}_k . U metod nepoužívajících Josephův tvar se do měřeného času navíc započítává i doba nutné symetrizace podle rovnice (2.10). Dále pak u sekvenčních přístupů není započítána de Korelace šumu měření, tzn. předpokládá se diagonální kovarianční matice šumu měření \mathbf{R}_k .

V měření je zahrnuta i doba výpočtu odhadu vektoru měření $\mathbf{h}(\hat{\mathbf{x}}_k^-)$. Způsob tohoto výpočtu závisí na modelu implementovaného KF. Při měření byla předpokládána lineární funkce \mathbf{h} a odhad vektoru měření byl počítán jako $\mathbf{H}\hat{\mathbf{x}}_k$. V případě nelineární funkce \mathbf{h} by se změřené časy lišili, nicméně u všech metod přibližně o stejný přírůstek.

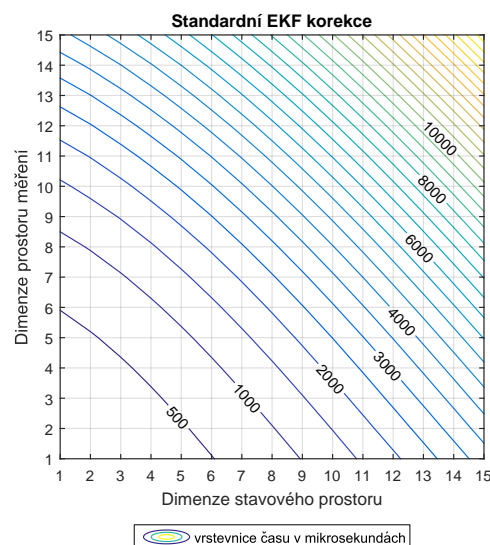
Měření probíhalo nad náhodně generovanými daty. Matice \mathbf{R} byla generována jako diagonální. Matici \mathbf{P}^0 byla následovně zajištěna symetričnost a pozitivní definitnost:

$$\mathbf{P}^0 = (\mathbf{P}_{gen} + \mathbf{P}_{gen}^T) + \mathbf{I} \cdot 50$$

Prvky \mathbf{P}_{gen} jsou z intervalu $\langle 0, 1 \rangle$. Maximální měřená dimenze \mathcal{S} i \mathcal{M} je 15, tedy matice \mathbf{P}_{gen} je nejvýše prvkem $\mathbb{R}^{15 \times 15}$. Výraz $\mathbf{P}_{gen} + \mathbf{P}_{gen}^T$ zajišťuje symetrii a přičtení 50 k prvkům na diagonále zajistí, že je matice \mathbf{P}^- striktně diagonálně dominantní, tj. součet všech nediagonálních prvků v jednom řádku je menší než diagonální prvek daného řádku. Symetrická, striktně diagonálně dominantní matice s reálnými pozitivními čísly na diagonále je pozitivně definitní.

Standardní korekce

Na obrázku 5.1 je vidět časová náročnost korekčního kroku původního programu.



Obrázek 5.1: Původní standardní korekce s původní implementací maticových operací

Jak bylo popsáno výše, model KF, který je počítán programem navigačního systému, operuje s 12-ti prvkovým vektorem stavů a 4 prvkovým nebo 1 prvkovým vektorem měření. Pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 4$ trvá korekční krok $3043 \mu\text{s}$ a pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 1$ trvá $1904 \mu\text{s}$.

Dále v textu jsou pro každou metodu zobrazeny dva grafy. Vlevo je znázorněna časová náročnost při použití původní implementace maticových operací a vpravo při použití DSP knihovny.

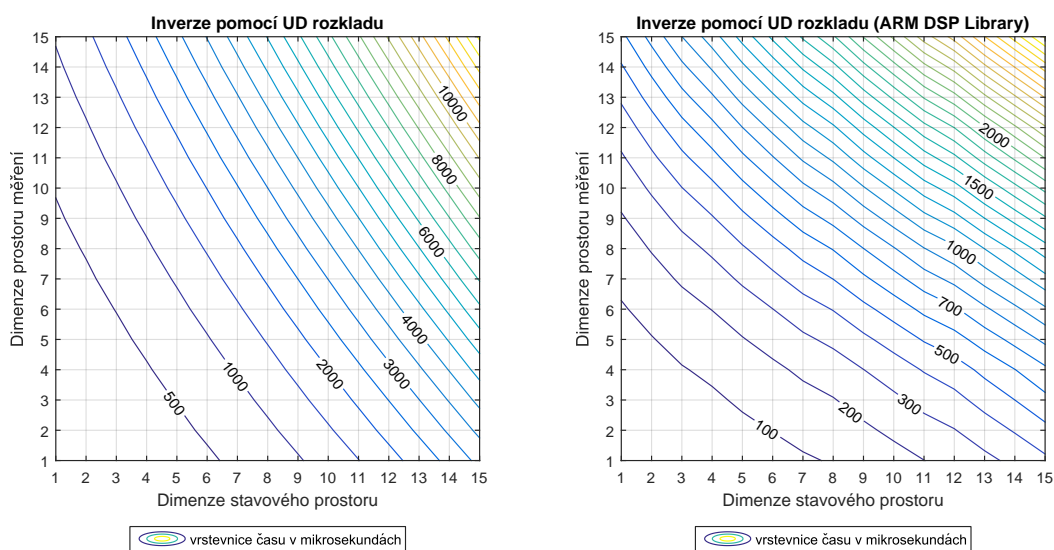
Inverze pomocí UD rozkladu

Na obrázku 5.2a je časová náročnost při výpočtu inverzní matice v rovnici 2.7 pomocí UD dekompozice s původní implementací maticových operací. Tvar vrstevnic je na první pohled odlišný od standardní metody - časová náročnost neroste tak rychle se zvětšující se dimenzí prostoru měření (\mathcal{M}). Pro počet prvků stav. vektoru 12 a počet prvků vektoru měření 4 ($\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 4$) je časová náročnost $2890 \mu\text{s}$. Pro tyto konkrétní dimenze prostorů měření a stavů je tedy tato metoda oproti předchozí pouze o zhruba $0,15 \text{ ms}$ rychlejší. Větší rychlost této metody se projeví u větších dimenzí prostoru měření.

Zajímavější je pohled vpravo na obrázek 5.2b. Stejný algoritmus je při použití DSP knihovny průměrně 4,1krát rychlejší než při použití původní implementace maticových operací. Konkrétně pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 4$ je časová náročnost UD korekce s DSP knihovnou $452 \mu\text{s}$, tedy je více než 6,3krát rychlejší než UD korekce bez DSP knihovny.

Sekvenční korekce

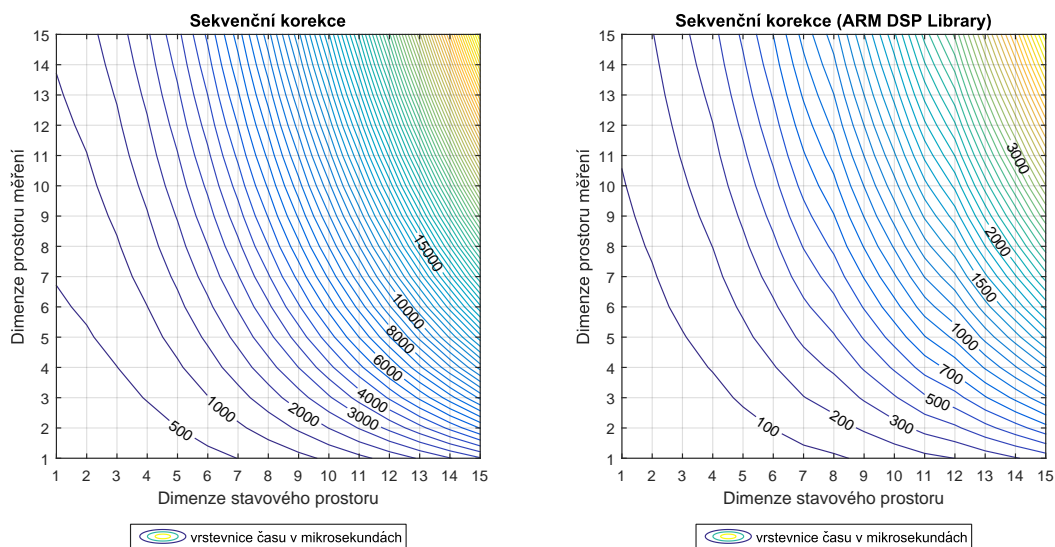
Na obrázku 5.3 jsou grafy pro sekvenční korekci. Při porovnání sekvenční korekce s předchozí metodou ve variantě s použitím původní implementace maticových operací není



(a) s původní implementací maticových operací

(b) s ARM DSP Library

Obrázek 5.2: Inverze pomocí UD rozkladu



(a) s původní implementací maticových operací

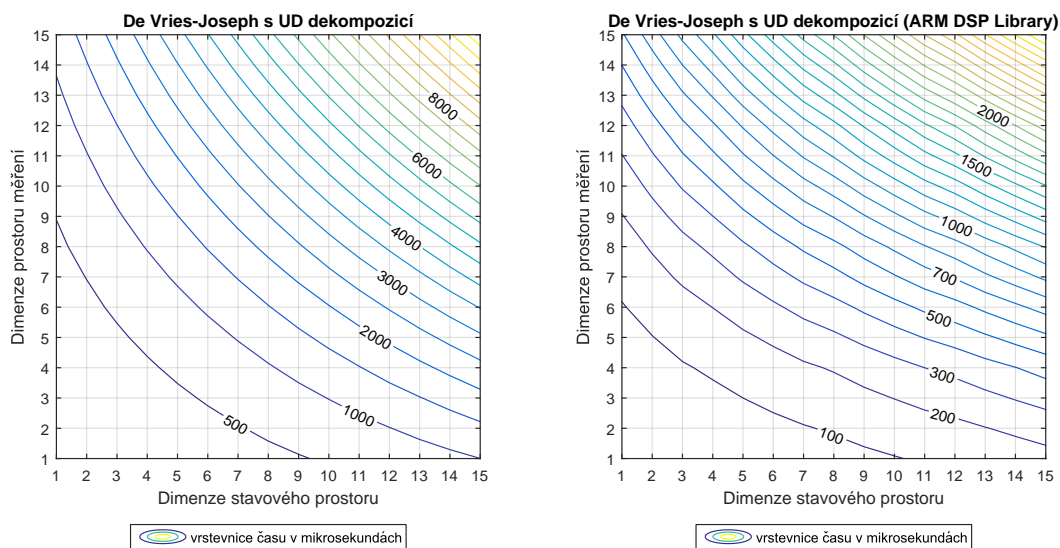
(b) s ARM DSP Library

Obrázek 5.3: Sekvenční korekce

vidět žádná velká výhoda sekvenčního přístupu. Při porovnání těchto metod ve variantě s DSP knihovnou je vidět výhoda sekvenční korekce při vyšších dimenzích prostoru měření. S rostoucí dimenzí stavového prostoru roste časová náročnost nejrychleji ze všech metod.

De Vries-Joseph s UD dekompozicí

Na obrázku 5.4 je znázorněna časová náročnost De Vriesovy metody s výpočtem Kalmanova zesílení pomocí UD rozkladu. Tato metoda poskytuje výhodu oproti předchozím



(a) s původní implementací maticových operací

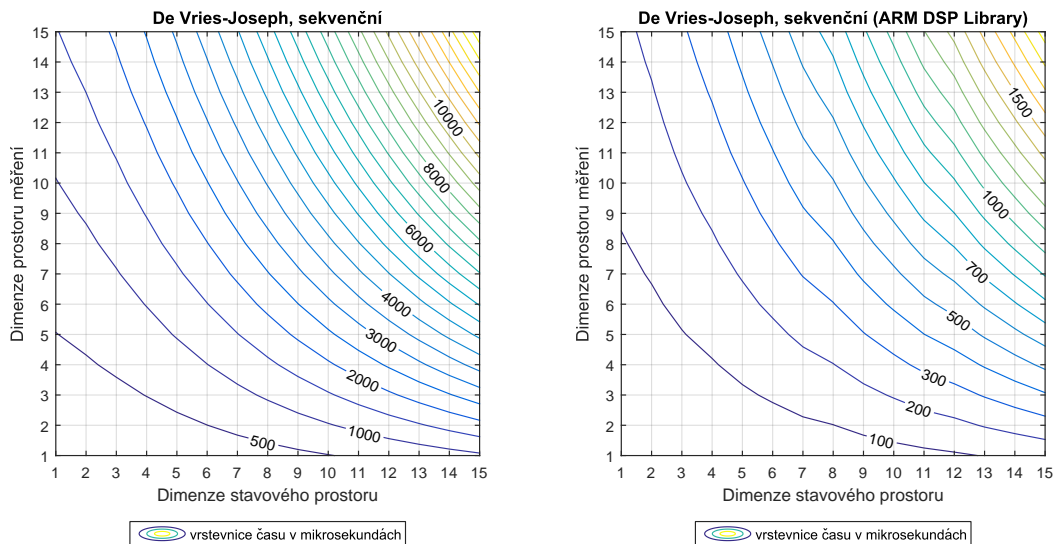
(b) s ARM DSP Library

Obrázek 5.4: De Vries-Joseph s UD dekompozicí

při vyšších dimenzích stavového vektoru.

De Vries-Joseph, sekvenční

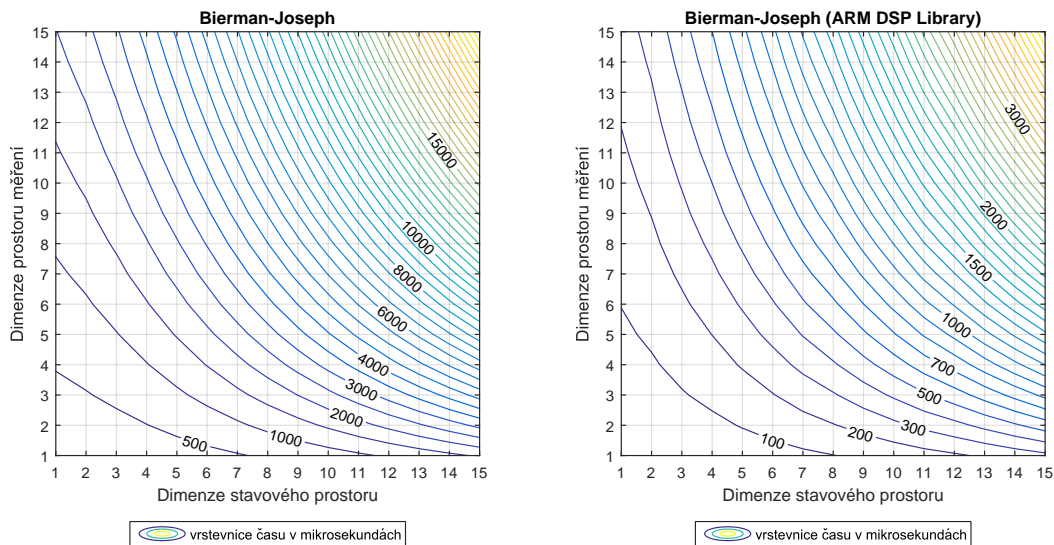
Na obrázku 5.5 je druhá z variant De Vriese - sekvenční. Při použití DSP knihovny je sekvenční varianta De Vrieseovy metody převážně rychlejší než UD varianta. Při použití původní implementace maticových operací je vhodnější využít UD variantu.



(a) s původní implementací maticových operací

(b) s ARM DSP Library

Obrázek 5.5: De Vries-Joseph, sekvenční



(a) s původní implementací maticových operací

(b) s ARM DSP Library

Obrázek 5.6: Bierman-Joseph

Bierman-Joseph

Grafy na obrázku 5.6 patří Biermanově metodě. Tato metoda při porovnávání s ostatními nijak nevynikla. Stejně jako u sekvenční metody roste rychle časová náročnost s dimenzí stavového prostoru.

5.2.2 Shrnutí

Měření ukázalo, že se vyplatí použití DSP knihovny namísto původních maticových operací. V tabulce 5.1 je vypsáno, kolikrát je metoda korekce KF při použití DSP knihovny průměrně rychlejší oproti té samé metodě při použití maticových operací původních. Použití DSP knihovny nejvíce urychlilo sekvenční metody, nicméně i u ostatních metod je zrychlení nezanedbatelné.

Metoda	Prům. rychlejší
inverze pomocí UD dekompozice	4.2026×
sekvenční korekce	7.2882×
De Vries-Joseph s UD dekompozicí	3.8893×
De Vries-Joseph, sekvenční	7.1645×
Bierman-Joseph	6.0285×

Tabulka 5.1: Kolikrát je metoda korekce s DSP knihovnou průměrně rychlejší než bez ní?

Program navigační jednotky počítá KF pro 12-ti prvkový stavový vektor a přepíná se mezi 1 a 4 prvkovým vektorem měření podle toho, zda jsou k dispozici data z GPS. V tabulce 5.2 jsou vypsány časy běhu korekčních metod pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 4$, a v tabulce 5.3 jsou časy korekce pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 1$. Z tabulek je vidět, že

Metoda	DSP knihovna	matrix_operation.c
původní korekce	-	3043 μs
inverze pomocí UD dekompozice	452 μs	2890 μs
sekvenční korekce	742 μs	6019 μs
De Vries-Joseph s UD dekompozicí	327 μs	1682 μs
De Vries-Joseph, sekvenční	356 μs	2562 μs
Bierman-Joseph	740 μs	4261 μs

Tabulka 5.2: Časová náročnost pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 4$

konkrétně pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 4$ se nejvíce vyplatí De Vriesova metoda s UD dekompozicí s použitím DSP knihovny. Jejím použitím klesne doba výpočtu Kalmanova zesílení \mathbf{K} , korekce \mathbf{P} , korekce $\hat{\mathbf{x}}$ a zajištění symetrie \mathbf{P} z původních 3043 μs na 327 μs . Pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 1$ je nejvýhodnější použití sekvenční De Vriesovy metody s DSP knihovnou. Čas korekce díky ní klesne z 1904 μs na 90 μs

Na obrázku 5.7 je znázorněno, která z metod je nejméně časově náročná pro dané dimenze prostoru měření a stavového prostoru při použití DSP knihovny a bez ní. Grafy byly vytvořené skriptem `compare.m`, viz CD. Konkrétní časy a zvětšené grafy jsou v příloze

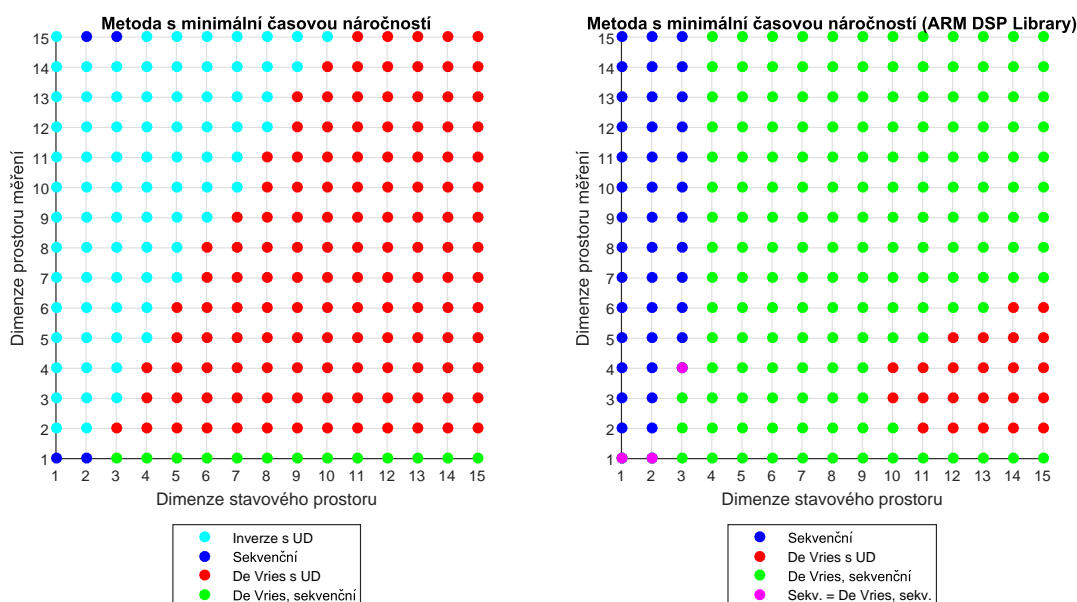
Metoda	DSP knihovna	matrix_operation.c
původní korekce	-	1904 μs
inverze pomocí UD dekompozice	228 μs	1822 μs
sekvenční korekce	201 μs	1687 μs
De Vries-Joseph s UD dekompozicí	121 μs	707 μs
De Vries-Joseph, sekvenční	90 μs	642 μs
Bierman-Joseph	186 μs	1066 μs

Tabulka 5.3: Časová náročnost pro $\dim \mathcal{S} = 12$ a $\dim \mathcal{M} = 1$

B.1.7 a B.1.8. Při použití DSP knihovny jsou většinou výhodnější metody založené na sekvenčním přístupu. Bez ní jsou naopak dominantní metody, které ve výpočtu Kalmanova zesílení využívají UD dekompozici.

Standardní způsob korekce implementovaný v neoptimalizovaném programu není nejvýhodnější pro žádnou kombinaci $\dim \mathcal{S}$ a $\dim \mathcal{M}$. Stejně tak Biermanova metoda se neukázala být nejrychlejší v žádném z měřených případů.

Všechny implementované metody korekce byly vyzkoušeny nad částí dat ze souboru 20141031_Flight_01_DATA.mat. V tabulce 5.4 je pro každou metodu korekce vypsán průměrný čas jednoho cyklu KF nad těmito daty bez použití jakýchkoli optimalizací původního kódu a bez DSP knihovny. Tzn. časové rozdíly oproti původnímu programu jsou způsobeny pouze jinou korekční metodou. Dále je v tabulce pro každou metodu uvedena maximální odchylka posledního stavu odhadnutého touto metodou od posledního stavu odhadnutého původním programem.



(a) s původní implementací maticových operací

(b) s ARM DSP Library

Obrázek 5.7: Metoda s minimální časovou náročností

Metoda	Prům. čas	Max. odchylka
původní korekce	9631 μs	-
inverze pomocí UD dekompozice	9539 μs	$8,6 \times 10^{-6}$
sekvenční korekce	10989 μs	$3,7 \times 10^{-6}$
De Vries-Joseph s UD dekompozicí	8344 μs	$4,1 \times 10^{-6}$
De Vries-Joseph, sekvenční	8771 μs	$1,7 \times 10^{-5}$
Bierman-Joseph	9803 μs	$3,4 \times 10^{-6}$
De Vries, kombinace UD a sekv.	8311 μs	$4,1 \times 10^{-6}$

Tabulka 5.4: Průměrná rychlost jednoho cyklu KF nad testovacími daty

5.3 Predikce \mathbf{P}_k pomocí operací se submaticemi

Zde je měřena efektivita způsobu predikce popsaného v kapitole 4.3. Testovací program pro porovnání efektivit klasické predikce s predikcí pomocí operací se submaticemi je v souboru `hypermatrix_test.c`.

5.3.1 Měření

Testovací program po spuštění náhodně vygeneruje čtvercové matice \mathbf{P} , \mathbf{F} a \mathbf{Q} o velikosti, která je daná hodnotou makra `STATE_VEC_SIZE` (v `main.h`), které se rovná dimenzi stavového prostoru. Symetrie náhodného \mathbf{P} je zajištěna spočítáním $\mathbf{P} = \mathbf{P}_{gen} + \mathbf{P}_{gen}^T$. Do vygenerované matice matic \mathbf{F} byly dále doplněny ukazatele na nulové a jednotkové submatice (při použití DSP knihovny), resp. jejich kódy (1 = jednotková, 2 = nulová) do pole `info` (při použití funkcí z `matrix_operation.c`), tak aby struktura odpovídala (4.1), tzn. tak, aby struktura odpovídala matici \mathbf{F} z modelu KF počítaného optimalizovanou navigační jednotkou.

Definicí makra `STANDARD_PREDICTION` se přepíná mezi použitím standardní predikce a predikce s použitím operací se submaticemi.

Porovnání metod predikce pro původní \mathbf{F}

Nejdříve byla porovnána efektivita klasických maticových operací a operací se submaticemi pro konkrétní velikost a strukturu \mathbf{F} z původního programu, tzn. matice 12×12 jdoucí rozdělit na matice matic podle (4.1), tedy

$$\mathbf{F}_k = \begin{bmatrix} \mathbf{F}_k^{1,1} & \mathbf{F}_k^{1,2} \\ \mathbf{0}_{6 \times 6} & \mathbf{F}_k^{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{F}_k^{1,2} & \mathbf{F}_k^{1,3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{F}_k^{2,2} & \mathbf{F}_k^{2,3} & \mathbf{F}_k^{2,4} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{F}_k^{3,3} & \mathbf{F}_k^{3,4} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix}$$

V tabulce 5.5 jsou naměřené časy predikce \mathbf{P} (v závorce jsou velikosti submatic, pomocí kterých je \mathbf{F} reprezentována). Porovnávány byly časy predikce \mathbf{P} při reprezentaci matice \mathbf{F} jako normální matice, matice matic se submaticemi o velikosti 6×6 a matice matic se submaticemi o velikosti 3×3 .

Metoda	DSP knihovna	matrix_operation.c
klasické mat. operace	237 μs	2300 μs
submat. operace (6×6)	340 μs	2317 μs
submat. operace (3×3)	442 μs	2528 μs

Tabulka 5.5: Porovnání efektivity predikce \mathbf{P}

Pro $\dim \mathcal{S} = 12$ a strukturu matice \mathbf{F} podle (4.1) je tedy při použití DSP knihovny jasně nejvýhodnější použít klasické maticové operace. Při použití maticových operací ze souboru `matrix_operation.c` mají klasické operace obdobnou časovou náročnost jako operace se submaticemi z $\mathbb{R}^{6 \times 6}$.

Různé dimenze, $1/4$ prvků matice \mathbf{F} je nulová

Rychlost výpočtu pomocí operací se submaticemi velmi záleží na struktuře \mathbf{F} a na tom, jakým způsobem se matice rozdělí. Možných kombinací struktur a rozdělení je velké množství. Dále je změřena jedna z nich. Matice \mathbf{F} je rozdělena na čtyři submatice, z nichž jedna je nulová. V předchozím měření byla měřena tato kombinace pro $\dim \mathcal{S} = 12$. Při použití DSP knihovny byl výpočet operací se submaticemi o 103 μs pomalejší než výpočet s klasickými operacemi.

Cílem tohoto měření je zjistit pro jakou dimenzi stavového prostoru by bylo již výhodnější použít operace se submaticemi při této kombinaci struktury \mathbf{F} a jejího rozdělení. Porovnání je na obrázku 5.8. Naměřené hodnoty jsou označeny křížkem. Body byly proloženy polynomem 3. stupně metodou nejmenších čtverců při fixaci průchodu křivky počátkem souřadnic. Graf byl vytvořen pomocí Matlabu, viz `prediction.m` na příloženém CD.

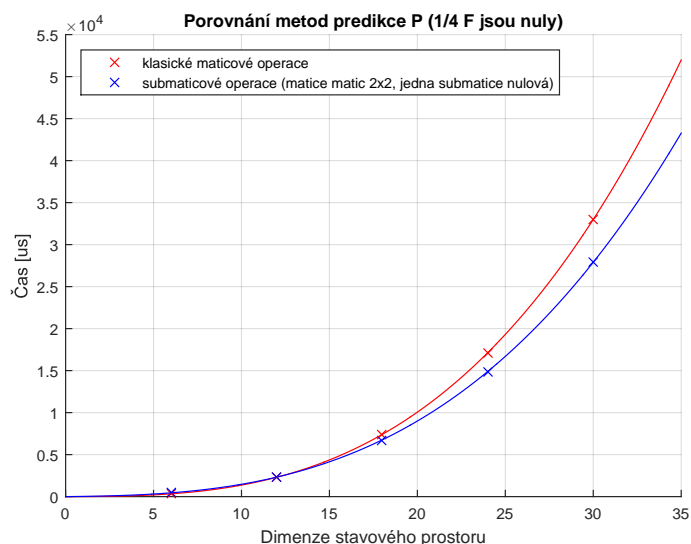
Při použití DSP knihovny začínají být operace se submaticemi rychlejší přibližně od $\dim \mathcal{S} = 25$. Poté časová náročnost predikce s klasickými maticovými operacemi začíná růst výrazněji. Bez DSP knihovny jsou operace se submaticemi v tomto případě rychlejší už zhruba od $\dim \mathcal{S} = 15$.

5.3.2 Shrnutí

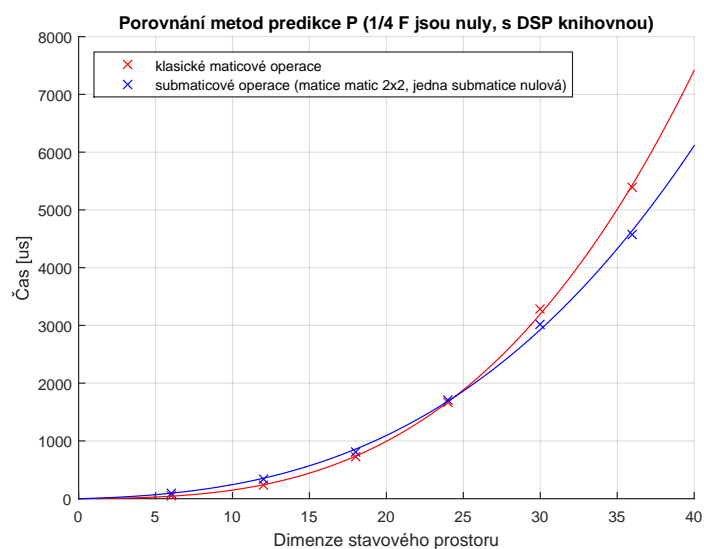
Měření ukázalo, že stávající program by se použitím operací se submaticemi zpomalil, byť v případě submatic o velikosti 6×6 jen o pár mikrosekund. Značného urychlení se ale dosáhne použitím DSP knihovny, a to přímo jejích klasických maticových operací a ne nově implementovaných operací se submaticemi. Predikce \mathbf{P} se použitím DSP knihovny urychlí více než 9,7krát z původních 2300 μs na 237 μs .

Časová náročnost varianty se submaticovými operacemi závisí nejen na dimenzi vektoru měření, ale i na struktuře matice \mathbf{F} a jejím rozdělení na submatice. Konkrétně při rozdělení na čtyři matice, kde jedna z nich je nulová se nově implementovaná metoda predikce \mathbf{P} při použití DSP knihovny vyplatí až při $\dim \mathcal{S} = 25$ a při použití maticových operací z `matrix_operation.c` od zhruba $\dim \mathcal{S} = 15$.

Problém při použití DSP knihovny je, že před tím, než se spustí samotné násobení se submaticemi, je potřeba převést klasickou matici \mathbf{P} do podoby matice matic reprezen-



(a) s původní implementací maticových operací



(b) s ARM DSP Library

Obrázek 5.8: Porovnání způsobů predikce

tovanou strukturou *hypermatrix*. Klasická matice je uložena v paměti po řádcích, pro správnou funkci operací se submaticemi je potřeba mít matici v paměti uloženou po jednotlivých submaticích. Nutnost převodu pak celou predikci matice \mathbf{P} zpomaluje.

5.4 Předpočítávání

Jak bylo uvedeno v kapitole „Optimalizace“, největší úsporu času pomocí předpočítání přineslo předpočítání goniometrických funkcí namísto jejich vícenásobného volání pro stejné hodnoty. Pro příklad funkce `compute_process_model_C_code_ready`, která aktualizuje stav dynamického systému a matici \mathbf{F}_k , trvala v neoptimalizovaném programu 1863 μs . Má-li funkce k dispozici předpočítané hodnoty, doba běhu klesne na 308 μs .

6 Závěr

Cílem bakalářské práce bylo urychlit výpočet algoritmu Kalmanova filtru na navigační jednotce s mikroprocesorem ARM Cortex-M4. V rámci bakalářské práce byly popsány, implementovány a porovnány různé metody a způsoby implementace korekčního kroku Kalmanova filtru. Dále byl navržen způsob predikce kovarianční matice \mathbf{P} s využitím operací se submaticemi. Původní software navigační jednotky byl upraven tak, aby nově umožňoval přepnutí do režimu používání maticových operací z CMSIS DSP knihovny. Dále byl software upraven tak, aby pokud možno využíval předpočítaných mezivýsledků.

Pro daný model Kalmanova filtru z programu navigační jednotky (dvanáctiprvkový stavový vektor a jednoprvkový či čtyřprvkový vektor měření) se z implementovaných metod korekcí nejvíce hodí kombinace De Vriesovy sekvenční metody a De Vriesovy metody s UD dekompozicí, které se přepínají v závislosti na tom, zda vektor měření obsahuje pozici z GPS (má čtyři prvky) nebo ne (má jeden prvek). Pro daný model Kalmanova filtru se neukázalo být výhodné použití navržené implementace predikce \mathbf{P} pomocí operací se submaticemi. Použití CMSIS DSP knihovny a předpočítání mezivýsledků urychluje výpočet nezávisle na modelu Kalmanova filtru.

Průměrná doba jednoho cyklu Kalmanova filtru v původním neoptimalizovaném programu nad daty z `20141031_Flight_01_DATA.mat` byla $9631 \mu\text{s}$. Po optimalizaci klesla průměrná doba výpočtu jednoho cyklu KF na $1206 \mu\text{s}$. Výpočet algoritmu KF je tedy nad těmito daty po optimalizaci téměř osmkrát rychlejší než před ní. Konkrétně nad těmito daty se poslední stav odhadnutý optimalizovaným programem lišil od posledního stavu odhadnutého programem původním maximálně o méně než $4,1 \times 10^{-6}$.

Seznam použité literatury

- [1] CAI, G., CHEN, B. M. a LEE, T. H. *Unmanned Rotorcraft Systems*, kapitola Coordinate Systems and Transformations. Springer-Verlag London, 2011. ISBN 978-0-85729-634-4.
- [2] FADALI, M. S. *Sequential DKF Computation* [online]. [cit. 30. 3. 2016]. Dostupné z: <<http://wolfweb.unr.edu/~fadali/EE782/OneAtATimeDKF.pdf>>.
- [3] GREWAL, M. S. a ANDREWS, A. P. *Kalman filtering: theory and practice using MATLAB*. Wiley, 2008, 3. edice. ISBN 978-0-470-37781-9.
- [4] GROVES, P. D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008. ISBN 978-1-58053-255-6.
- [5] *MPU-9150 Product Specification* [online]. InvenSense, rev. 4.3. Dostupné z: <<http://43zrtwysvxb2gf29r5o0athu.wpengine.netdna-cdn.com/wp-content/uploads/2015/02/MPU-9150-Datasheet.pdf>>.
- [6] *Its Definition and Relationships with Local Geodetic Systems* [online]. National Imagery and Mapping Agency, Department of Defense World Geodetic System 1984. [cit. 20. 4. 2016]. Dostupné z: <<http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>>.
- [7] SALIH, A. A. A.-A., ZAINI, N. L. A. C. A. a ZHAHIR, A. The suitability of GPS receivers update rates for navigation applications. *International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering* [online], 2013, **7** (6). Dostupné z: <<http://waset.org/publications/7783>>.
- [8] SOTÁK, M., SOPATA, M., BRÉDA, R., ROHÁČ, J. a VÁCL, L. *Integrácia navigačných systémov*. Košice: Róbert Bréda „Vydané nákladom vlastným“, 2006. ISBN 80-969619-9-3.
- [9] *RM0090 Reference manual* [online]. STMicroelectronics, rev. 11. Dostupné z: <http://www2.st.com/content/ccc/resource/technical/document/reference_manual/3d/6d/5a/66/b4/99/40/d4/DM00031020.pdf/files/DM00031020.pdf/jcr:content/translations/en.DM00031020.pdf>.
- [10] *MAX-M8, u-blox M8 Concurrent GNSS modules, Data Sheet* [online]. u-blox, rev. 10. Dostupné z: <https://www.u-blox.com/sites/default/files/MAX-M8_DataSheet_%28UBX-13004644%29.pdf>.

Seznam zkratek

API	Application Programming Interface
CMSIS	Cortex Microcontroller Software Interface Standard
DSP	Digital signal processing
ECEF	Earth-centered, Earth-fixed - souřadnicový systém
ECI	Earth-centered inertial - souřadnicový systém
EKF	rozšířený Kalmanův filtr
FOG	Fiber Optic Gyro - typ senzoru úhlové rychlosti
FPU	Floating Point Unit - matematický koprocessor
GNSS	globální navigační satelitní systém
GPS	globální poziční systém
IMU	inerciální měřicí jednotka
INS	inerciální navigační systém
KF	Kalmanův filtr
MEMS	Micro-Electro-Mechanical Systems
NED	North-East-Down - souřadnicový systém
RLG	Ring Laser Gyro - typ senzoru úhlové rychlosti
SIMD	Single Instruction, Multiple Data
WGS 84	World Geodetic System 1984

Seznam symbolů

\mathbf{x}	stavový vektor
$\hat{\mathbf{x}}^-$	odhad stavového vektoru po predikci
$\hat{\mathbf{x}}^+$	odhad stavového vektoru po korekci
\mathbf{y}	vektor měření
\mathbf{P}	kovarianční matice odhadu stavového vektoru
\mathbf{P}^-	kovarianční matice odhadu stavového vektoru po predikci
\mathbf{P}^+	kovarianční matice odhadu stavového vektoru po korekci
\mathbf{R}	kovarianční matice šumu měření
\mathbf{Q}	kovarianční matice procesního šumu
\mathbf{K}	matice Kalmanova zesílení
\mathbf{f}	specifická síla nebo funkce určující dynamický systém
\mathbf{h}	řádek matice \mathbf{H} nebo funkce určující vztah vektoru měření se stavovým vektorem
\mathbf{F}	Jakobián funkce \mathbf{f}
\mathbf{H}	Jakobián funkce \mathbf{h}
$\boldsymbol{\omega}_{ib}^b$	úhlová rychlost vztažné soustavy navigovaného tělesa vůči inerciální vztažné soustavě vyjádřená v souřadnicovém systému navigovaného tělesa
\mathbf{f}_{ib}^b	specifická síla vztažné soustavy navigovaného tělesa vůči inerciální vztažné soustavě vyjádřená v souřadnicovém systému navigovaného tělesa
\mathbf{a}_{ib}	zrychlení vztažné soustavy navigovaného tělesa vůči inerciální vztažné soustavě
\mathbf{p}	vektor pozice
\mathbf{v}	vektor rychlosti
\mathcal{S}	stavový prostor
\mathcal{M}	prostor měření

A CD

```
root
├── BP_Martin_Paclik_2016.pdf .....Tento text ve formátu PDF.
├── C .....Kódy navigační jednotky v jazyce C.
│   ├── new .....Kódy nově vzniklé jako součást této BP.
│   │   ├── library_switch.h
│   │   ├── correction.c
│   │   ├── correction.h
│   │   ├── correction_test.c
│   │   ├── hypermatrix_test.c
│   │   ├── arm_mat.c
│   │   ├── arm_mat.h
│   │   ├── hypermatrix.c
│   │   ├── hypermatrix.h
│   │   ├── timer5.c
│   │   ├── timer5.h
│   │   ├── original.c
│   │   └── original.h
│   ├── modified .... Kódy navigační jednotky, které byly v rámci této BP upraveny.
│   │   ├── kalman_my.c
│   │   ├── main.c
│   │   └── main.h
│   ├── original_unmodified Kódy, které jsou potřeba pro správnou funkci navigační
│   │                       jednotky, ale které nebyli v rámci této BP nijak upravo-
│   │                       vány a zůstali v původní podobě.
│   │   ├── matrix_operation.c
│   │   ├── matrix_operation.h
│   │   ├── NED2UTM.c
│   │   ├── NED2UTM.h
│   │   ├── IO_cntrl.h
│   │   └── kalman_my.h
│   └── Matlab .....Kódy v Matlabu pro vykreslení grafů použitých v BP.
│       ├── compare.m
│       ├── data_arm.m
│       ├── data_noarm.m
│       ├── graph.m
│       └── prediction.m
└── data ..... Data z letu, na kterých byl KF mimo jiné testován.
    └── 20141031_Flight_01_DATA.mat
```

B Naměřené hodnoty a grafy

B.1 Metody korekce

B.1.1 Standardní EKF

		Dimenze stavového prostoru														
	[μ s]	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dimenze prostoru měření	1	160	187	229	291	375	487	628	803	1015	1266	1562	1904	2297	2743	3247
	2	186	223	280	360	466	602	772	979	1225	1515	1853	2240	2680	3179	3738
	3	229	280	353	453	583	746	947	1187	1471	1801	2183	2618	3109	3661	4279
	4	294	361	453	576	732	925	1157	1433	1757	2131	2558	3043	3587	4196	4873
	5	387	471	585	733	916	1141	1409	1724	2089	2507	2983	3519	4121	4788	5526
	6	512	616	753	928	1143	1401	1706	2062	2470	2937	3464	4054	4712	5440	6243
	7	673	800	964	1167	1415	1709	2054	2452	2907	3423	4003	4650	5367	6159	7029
	8	876	1028	1220	1455	1737	2069	2456	2899	3403	3971	4606	5312	6092	6950	7888
	9	1125	1304	1526	1795	2115	2489	2919	3409	3964	4586	5279	6046	6890	7814	8825
	10	1426	1633	1888	2194	2553	2969	3445	3987	4594	5272	6025	6855	7766	8761	9844
	11	1781	2021	2311	2655	3055	3517	4042	4635	5298	6035	6849	7745	8725	9792	10951
	12	2198	2473	2799	3184	3629	4137	4713	5360	6080	6879	7758	8721	9772	10914	12150
	13	2681	2991	3357	3784	4275	4835	5464	6166	6947	7808	8754	9787	10912	12130	13447
	14	3234	3582	3991	4463	5002	5612	6297	7059	7902	8829	9843	10949	12149	13446	14846
	15	3861	4250	4702	5223	5813	6477	7219	8043	8949	9944	11030	12210	13488	14867	16351

Tabulka B.1: Standardní korekce

B.1.2 Inverze pomocí UD dekompozice

[μ s]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	128	153	193	252	335	443	580	749	955	1201	1488	1822	2206	2643	3135
2	146	182	237	316	420	553	718	919	1161	1445	1774	2154	2587	3076	3624
3	168	218	289	388	515	675	870	1106	1384	1708	2081	2508	2991	3535	4140
4	197	261	352	472	625	813	1041	1312	1629	1995	2415	2890	3426	4025	4690
5	232	315	425	569	750	969	1231	1539	1896	2308	2775	3302	3891	4548	5275
6	275	376	510	680	889	1141	1439	1787	2188	2645	3161	3742	4388	5105	5895
7	325	447	606	804	1044	1330	1667	2057	2502	3008	3577	4212	4917	5695	6551
8	383	528	713	941	1215	1539	1915	2348	2841	3396	4019	4711	5476	6319	7241
9	450	620	833	1093	1403	1764	2183	2662	3204	3811	4489	5241	6069	6978	7969
10	524	722	966	1259	1607	2009	2472	2998	3590	4253	4988	5801	6694	7669	8733
11	608	835	1111	1441	1827	2273	2782	3357	4003	4721	5517	6392	7351	8397	9533
12	701	959	1270	1638	2065	2556	3113	3739	4440	5217	6074	7014	8042	9160	10371
13	804	1095	1443	1851	2321	2858	3465	4145	4902	5740	6660	7668	8766	9958	11247
14	917	1243	1629	2079	2595	3181	3840	4575	5391	6291	7278	8354	9525	10793	12160
15	1040	1403	1830	2323	2887	3524	4237	5030	5907	6870	7925	9073	10318	11663	13112

Tabulka B.2: Inverze pomocí UD dekompozice

[μ s]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	19	26	34	42	56	71	89	107	134	164	199	228	275	324	380
2	28	38	51	63	82	102	127	149	183	220	262	296	352	408	473
3	38	53	70	87	112	138	169	195	237	280	330	371	435	500	573
4	52	72	95	116	148	180	218	250	299	350	407	452	526	598	681
5	70	96	127	155	195	236	282	322	380	440	508	562	646	730	823
6	92	127	166	202	250	299	356	404	474	544	623	686	782	877	984
7	120	164	212	257	315	376	443	501	582	664	755	828	937	1046	1166
8	152	206	264	318	389	460	538	605	699	793	896	979	1101	1223	1355
9	191	256	327	392	476	560	651	730	838	946	1065	1160	1299	1436	1584
10	236	314	398	475	572	670	777	867	991	1114	1248	1357	1512	1666	1832
11	288	381	478	569	681	793	915	1020	1161	1300	1451	1575	1748	1919	2104
12	347	453	566	670	799	926	1065	1183	1341	1497	1664	1801	1993	2181	2384
13	415	537	667	786	932	1078	1234	1369	1545	1720	1908	2061	2274	2482	2706
14	490	630	777	913	1078	1242	1416	1568	1765	1960	2168	2339	2574	2803	3047
15	574	732	899	1051	1237	1421	1616	1786	2005	2220	2449	2639	2896	3147	3415

Tabulka B.3: Inverze pomocí UD dekompozice (ARM DSP Library)

B.1.3 Sekvenční korekce

		Dimenze stavového prostoru														
	[μ s]	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dimenze prostoru měření	1	92	113	149	203	279	378	506	666	860	1093	1368	1687	2055	2474	2949
	2	163	201	265	361	497	680	916	1212	1574	2010	2528	3131	3828	4626	5531
	3	235	289	380	520	717	982	1326	1758	2289	2928	3687	4575	5602	6778	8114
	4	306	377	496	679	936	1285	1736	2305	3004	3846	4847	6019	7375	8930	10698
	5	377	465	611	836	1156	1586	2146	2851	3718	4764	6007	7463	9149	11082	13280
	6	449	552	728	995	1376	1889	2557	3397	4432	5682	7167	8907	10923	13234	15863
	7	521	640	843	1153	1595	2191	2967	3943	5146	6600	8327	10351	12696	15386	18445
	8	592	728	959	1312	1815	2493	3376	4490	5862	7517	9486	11794	14470	17538	21028
	9	664	815	1075	1470	2034	2795	3786	5036	6576	8435	10646	13239	16243	19690	23611
	10	735	904	1190	1628	2253	3098	4196	5582	7290	9354	11806	14683	18017	21842	26193
	11	807	991	1306	1787	2473	3400	4606	6129	8004	10271	12967	16127	19790	23994	28776
	12	878	1079	1422	1945	2692	3702	5016	6674	8719	11189	14126	17571	21564	26146	31358
	13	950	1167	1537	2104	2911	4004	5426	7221	9433	12107	15286	19015	23338	28299	33941
	14	1021	1255	1653	2262	3131	4307	5836	7767	10148	13025	16447	20459	25111	30450	36523
	15	1092	1343	1769	2421	3351	4609	6246	8314	10862	13943	17606	21903	26885	32602	39106

Tabulka B.4: Sekvenční korekce

		Dimenze stavového prostoru														
	[μ s]	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Dimenze prostoru měření	1	13	17	24	31	42	56	73	88	113	142	175	201	246	293	347
	2	22	30	42	56	77	102	135	164	212	265	330	382	467	558	663
	3	31	43	61	79	110	148	197	240	311	390	484	561	690	824	980
	4	40	56	78	104	145	194	258	316	410	514	639	742	911	1090	1296
	5	49	69	96	128	179	241	320	391	509	639	794	922	1133	1355	1613
	6	58	81	115	153	213	286	381	467	608	763	949	1102	1355	1621	1929
	7	68	94	133	177	247	333	444	543	706	887	1104	1282	1577	1888	2247
	8	77	107	151	201	281	379	505	620	806	1012	1259	1463	1799	2153	2563
	9	86	119	169	225	315	425	566	695	904	1137	1414	1643	2021	2418	2879
	10	95	132	187	250	349	471	629	771	1003	1261	1568	1823	2243	2684	3196
	11	104	145	206	274	383	517	690	847	1102	1385	1723	2003	2465	2950	3513
	12	113	157	223	298	417	563	751	923	1201	1510	1878	2184	2686	3216	3830
	13	123	170	241	322	451	609	813	998	1300	1634	2032	2363	2909	3482	4146
	14	132	182	260	347	485	656	875	1074	1399	1758	2188	2543	3130	3747	4463
	15	141	196	278	371	519	701	936	1150	1497	1883	2343	2724	3352	4013	4779

Tabulka B.5: Sekvenční korekce (ARM DSP Library)

B.1.4 De Vries-Joseph s UD dekompozicí

[μ s]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	137	156	183	216	254	299	351	410	474	545	623	707	797	894	998
2	156	189	230	279	338	405	481	566	659	761	871	990	1119	1255	1401
3	182	230	290	360	441	533	636	750	875	1011	1158	1315	1484	1663	1854
4	216	283	364	457	564	684	818	964	1123	1297	1482	1682	1895	2120	2359
5	258	346	452	572	707	859	1025	1207	1405	1618	1847	2091	2351	2626	2916
6	307	422	554	704	871	1057	1260	1481	1720	1976	2251	2542	2852	3180	3524
7	365	508	671	854	1057	1280	1522	1785	2069	2371	2693	3037	3400	3783	4185
8	433	607	804	1023	1263	1527	1813	2121	2450	2804	3178	3575	3994	4435	4899
9	509	719	952	1210	1492	1800	2131	2487	2867	3273	3703	4156	4635	5138	5665
10	596	843	1117	1417	1743	2097	2478	2885	3320	3780	4268	4782	5323	5892	6486
11	693	981	1297	1642	2017	2420	2853	3315	3806	4326	4874	5452	6060	6696	7360
12	800	1131	1494	1888	2313	2770	3258	3778	4328	4909	5523	6168	6844	7551	8289
13	917	1296	1708	2154	2634	3146	3692	4273	4886	5533	6214	6928	7676	8457	9271
14	1047	1476	1940	2440	2977	3548	4156	4801	5480	6195	6946	7734	8556	9415	10310
15	1188	1670	2190	2748	3344	3978	4651	5361	6111	6898	7722	8585	9486	10426	11403

Tabulka B.6: De Vries-Joseph s UD dekompozicí

[μ s]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	20	25	32	39	46	54	65	72	84	96	109	121	136	151	168
2	28	36	46	56	69	81	95	108	125	141	159	175	197	218	241
3	39	51	67	82	100	118	137	155	179	202	227	248	277	306	336
4	52	71	93	112	135	160	186	208	238	268	300	327	363	398	436
5	71	98	127	153	185	217	252	283	323	362	403	438	485	531	580
6	95	130	166	201	243	285	330	368	418	467	519	564	622	679	738
7	123	168	215	258	312	364	419	468	530	591	654	709	781	849	921
8	155	210	268	321	385	448	515	573	646	718	794	858	942	1022	1107
9	196	264	334	399	476	553	634	705	793	879	969	1047	1146	1242	1343
10	243	323	408	485	578	668	763	848	952	1053	1159	1250	1366	1478	1594
11	297	391	491	583	691	798	909	1008	1129	1247	1370	1476	1610	1740	1874
12	355	466	581	686	810	933	1059	1172	1309	1443	1583	1703	1853	1999	2150
13	424	553	686	808	951	1092	1238	1368	1525	1679	1838	1976	2147	2314	2486
14	501	648	800	939	1103	1262	1429	1577	1755	1928	2108	2265	2458	2645	2839
15	588	754	926	1085	1269	1450	1637	1804	2004	2200	2402	2578	2795	3005	3221

Tabulka B.7: De Vries-Joseph s UD dekompozicí (ARM DSP Library)

B.1.5 De Vries-Joseph, sekvenční

	Dimenze stavového prostoru															
$[\mu\text{s}]$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Dimenze prostoru měření	1	99	117	140	170	206	250	299	354	417	485	560	642	729	824	924
	2	198	232	279	340	412	498	597	707	832	969	1118	1282	1457	1646	1848
	3	296	347	419	508	617	746	894	1061	1247	1453	1678	1922	2185	2468	2770
	4	395	463	557	677	823	994	1191	1413	1662	1936	2236	2562	2913	3291	3692
	5	493	578	696	846	1028	1242	1488	1766	2077	2420	2795	3202	3641	4112	4615
	6	591	693	834	1014	1233	1490	1785	2120	2492	2904	3353	3842	4368	4934	5538
	7	689	809	974	1184	1438	1739	2083	2473	2908	3387	3912	4481	5097	5756	6461
	8	787	924	1112	1352	1643	1987	2381	2825	3322	3871	4471	5122	5825	6579	7384
	9	886	1040	1251	1522	1849	2235	2678	3178	3737	4355	5030	5762	6552	7401	8307
	10	984	1155	1390	1691	2054	2482	2975	3532	4152	4838	5588	6402	7280	8223	9229
	11	1082	1270	1529	1859	2259	2730	3272	3884	4568	5322	6147	7042	8008	9045	10152
	12	1181	1386	1668	2028	2465	2978	3569	4238	4983	5806	6705	7682	8736	9867	11075
	13	1279	1501	1807	2196	2670	3226	3867	4591	5399	6290	7264	8322	9464	10689	11998
	14	1377	1616	1946	2366	2875	3474	4165	4944	5814	6773	7822	8962	10192	11512	12921
	15	1475	1731	2084	2534	3080	3723	4462	5297	6229	7256	8381	9603	10920	12333	13844

Tabulka B.8: De Vries-Joseph, sekvenční

	Dimenze stavového prostoru															
$[\mu\text{s}]$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Dimenze prostoru měření	1	13	17	20	25	31	37	44	51	61	70	80	90	103	116	131
	2	25	31	40	48	60	73	88	99	119	139	160	178	206	232	261
	3	36	46	58	72	90	109	131	149	178	207	241	267	307	347	390
	4	49	60	78	95	119	145	174	198	237	277	320	356	409	462	520
	5	60	75	97	119	148	180	218	248	296	344	399	445	511	578	650
	6	72	90	116	142	178	217	261	297	355	414	479	533	613	692	780
	7	83	105	135	166	207	252	304	346	414	482	559	622	715	807	909
	8	95	119	154	190	236	288	347	395	473	551	638	710	817	923	1039
	9	107	134	174	213	266	324	391	444	532	619	718	799	919	1039	1169
	10	119	149	193	237	295	360	434	494	591	688	798	888	1021	1154	1298
	11	130	164	212	260	325	396	478	543	650	757	878	976	1122	1269	1428
	12	142	179	231	284	354	432	521	592	708	825	957	1064	1225	1384	1557
	13	153	194	251	308	384	468	564	642	768	894	1037	1153	1326	1499	1687
	14	166	209	269	331	413	504	607	691	826	963	1117	1242	1429	1615	1816
	15	177	223	288	354	443	540	651	740	886	1032	1196	1330	1530	1730	1947

Tabulka B.9: De Vries-Joseph, sekvenční (ARM DSP Library)

B.1.6 Bierman-Joseph

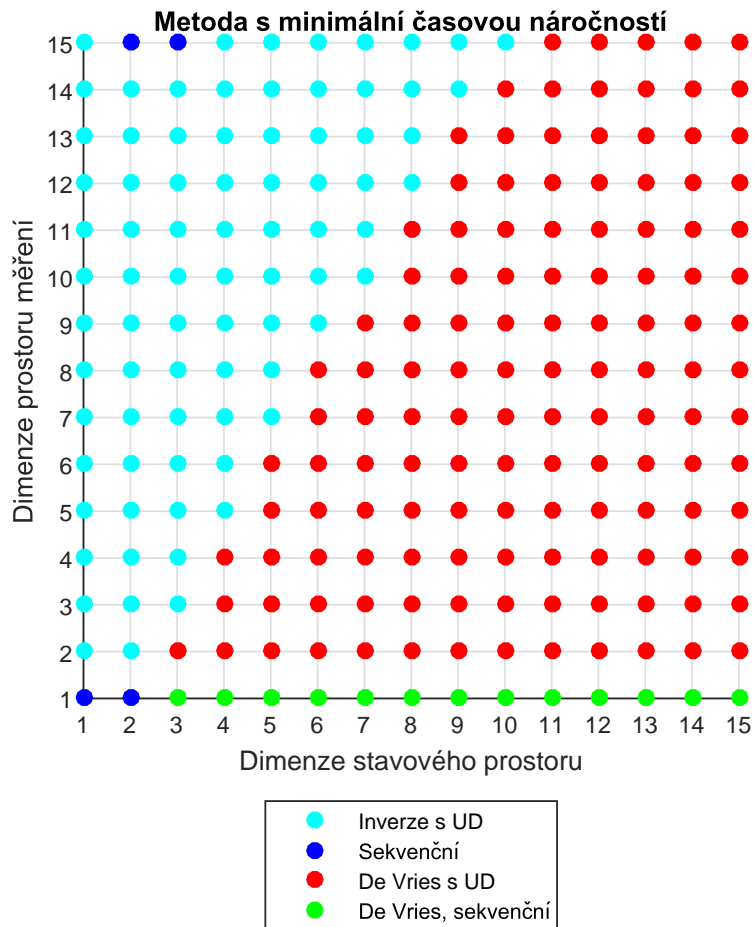
[μ s]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	133	159	197	246	307	381	466	562	670	790	922	1066	1222	1389	1568
2	266	317	392	491	614	759	928	1123	1339	1579	1844	2131	2442	2777	3135
3	398	476	587	736	920	1138	1393	1682	2008	2368	2763	3196	3663	4165	4703
4	528	634	783	981	1225	1518	1856	2243	2677	3158	3686	4261	4885	5552	6271
5	661	790	980	1227	1531	1898	2320	2805	3347	3947	4607	5324	6103	6940	7837
6	793	950	1175	1470	1837	2276	2785	3365	4015	4735	5528	6390	7325	8328	9404
7	925	1106	1370	1715	2145	2655	3248	3925	4684	5526	6450	7456	8545	9717	10970
8	1058	1266	1565	1960	2449	3035	3711	4487	5353	6315	7370	8522	9763	11104	12536
9	1190	1424	1763	2203	2756	3413	4176	5046	6022	7103	8292	9585	10984	12493	14102
10	1323	1580	1957	2448	3061	3792	4641	5606	6691	7894	9211	10649	12206	13880	15672
11	1453	1738	2153	2697	3368	4172	5104	6169	7359	8684	10133	11716	13427	15267	17239
12	1585	1895	2349	2940	3677	4552	5569	6726	8028	9472	11055	12778	14645	16654	18805
13	1719	2054	2544	3184	3980	4930	6036	7287	8697	10258	11975	13845	15867	18045	20373
14	1852	2213	2738	3430	4286	5309	6496	7847	9365	11048	12896	14908	17089	19429	21940
15	1983	2372	2933	3676	4591	5688	6959	8409	10035	11838	13818	15974	18306	20820	23506

Tabulka B.10: Bierman-Joseph

[μ s]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	18	23	32	41	53	67	82	98	117	139	163	186	214	245	276
2	35	46	63	81	105	132	163	195	234	277	325	371	427	487	551
3	52	68	94	121	156	196	244	291	351	415	486	556	640	730	827
4	68	91	124	161	208	261	325	388	467	552	648	740	854	973	1102
5	86	113	155	201	259	327	405	484	583	691	810	925	1067	1216	1376
6	102	135	185	241	311	391	487	581	699	827	971	1110	1279	1459	1651
7	119	157	216	281	362	456	567	678	816	966	1133	1295	1492	1702	1926
8	136	181	247	320	414	522	648	774	932	1103	1295	1480	1705	1945	2201
9	153	203	277	360	465	587	729	871	1049	1241	1456	1664	1918	2188	2476
10	169	225	307	400	517	651	810	967	1165	1379	1618	1849	2131	2431	2752
11	186	247	337	440	569	717	891	1064	1282	1517	1779	2034	2345	2673	3026
12	203	269	368	479	621	781	972	1161	1398	1655	1940	2219	2557	2916	3301
13	220	291	398	519	671	846	1053	1257	1514	1792	2103	2404	2771	3159	3577
14	236	314	429	559	723	911	1134	1354	1631	1930	2264	2588	2983	3402	3852
15	253	336	460	599	774	976	1214	1450	1747	2067	2426	2774	3197	3645	4126

Tabulka B.11: Bierman-Joseph (ARM DSP Library)

B.1.7 Metoda s nejmenší časovou náročností (bez ARM DSP Library)

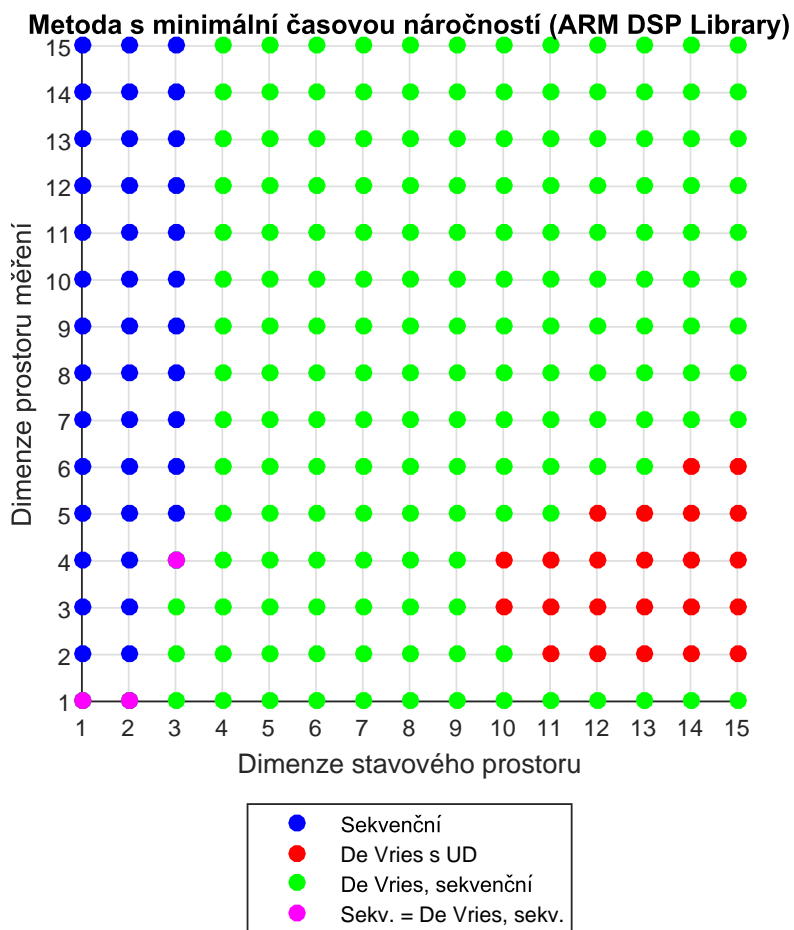


Obrázek B.1: Metoda s nejmenší časovou náročností (bez ARM DSP Library)

[μs]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	92	113	140	170	206	250	299	354	417	485	560	642	729	824	924
2	146	182	230	279	338	405	481	566	659	761	871	990	1119	1255	1401
3	168	218	289	360	441	533	636	750	875	1011	1158	1315	1484	1663	1854
4	197	261	352	457	564	684	818	964	1123	1297	1482	1682	1895	2120	2359
5	232	315	425	569	707	859	1025	1207	1405	1618	1847	2091	2351	2626	2916
6	275	376	510	680	871	1057	1260	1481	1720	1976	2251	2542	2852	3180	3524
7	325	447	606	804	1044	1280	1522	1785	2069	2371	2693	3037	3400	3783	4185
8	383	528	713	941	1215	1527	1813	2121	2450	2804	3178	3575	3994	4435	4899
9	450	620	833	1093	1403	1764	2131	2487	2867	3273	3703	4156	4635	5138	5665
10	524	722	966	1259	1607	2009	2472	2885	3320	3780	4268	4782	5323	5892	6486
11	608	835	1111	1441	1827	2273	2782	3315	3806	4326	4874	5452	6060	6696	7360
12	701	959	1270	1638	2065	2556	3113	3739	4328	4909	5523	6168	6844	7551	8289
13	804	1095	1443	1851	2321	2858	3465	4145	4886	5533	6214	6928	7676	8457	9271
14	917	1243	1629	2079	2595	3181	3840	4575	5391	6195	6946	7734	8556	9415	10310
15	1040	1343	1769	2323	2887	3524	4237	5030	5907	6870	7722	8585	9486	10426	11403

Tabulka B.12: Metoda s nejmenší časovou náročností (bez ARM DSP Library)

B.1.8 Metoda s nejmenší časovou náročností (s ARM DSP Library)



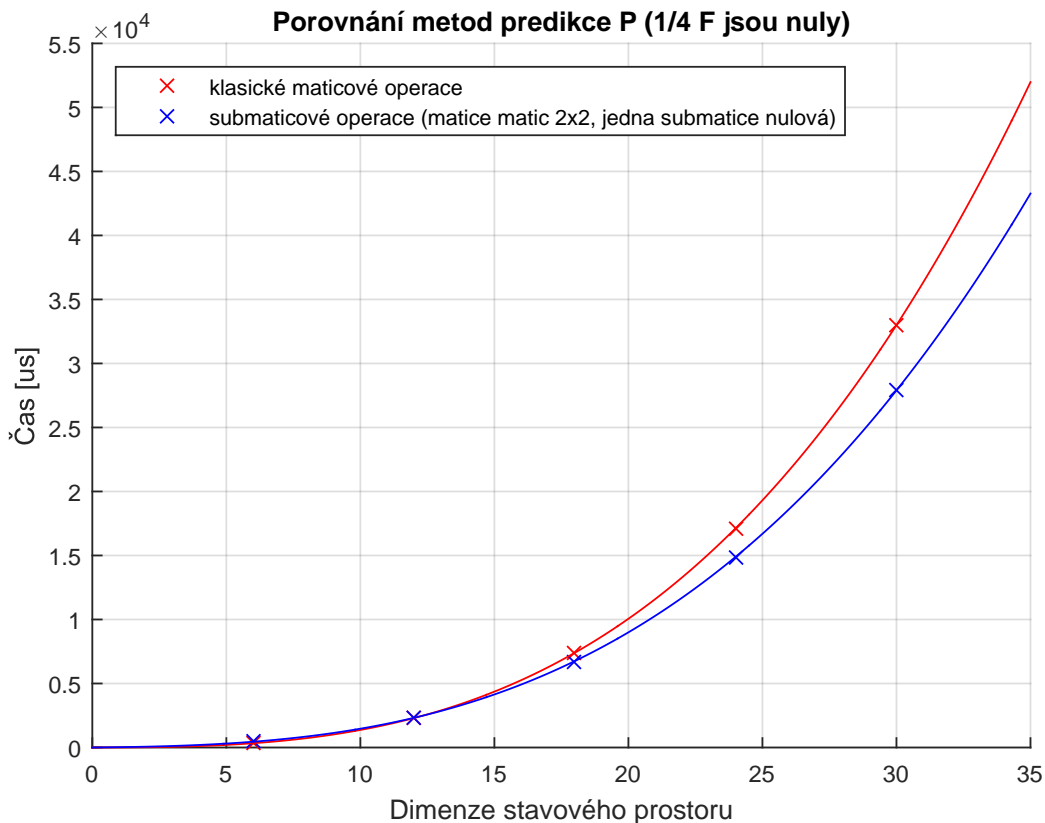
Obrázek B.2: Metoda s nejmenší časovou náročností (s ARM DSP Library)

[μs]	Dimenze stavového prostoru														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	13	17	20	25	31	37	44	51	61	70	80	90	103	116	131
2	22	30	40	48	60	73	88	99	119	139	159	175	197	218	241
3	31	43	58	72	90	109	131	149	178	202	227	248	277	306	336
4	40	56	78	95	119	145	174	198	237	268	300	327	363	398	436
5	49	69	96	119	148	180	218	248	296	344	399	438	485	531	580
6	58	81	115	142	178	217	261	297	355	414	479	533	613	679	738
7	68	94	133	166	207	252	304	346	414	482	559	622	715	807	909
8	77	107	151	190	236	288	347	395	473	551	638	710	817	923	1039
9	86	119	169	213	266	324	391	444	532	619	718	799	919	1039	1169
10	95	132	187	237	295	360	434	494	591	688	798	888	1021	1154	1298
11	104	145	206	260	325	396	478	543	650	757	878	976	1122	1269	1428
12	113	157	223	284	354	432	521	592	708	825	957	1064	1225	1384	1557
13	123	170	241	308	384	468	564	642	768	894	1037	1153	1326	1499	1687
14	132	182	260	331	413	504	607	691	826	963	1117	1242	1429	1615	1816
15	141	196	278	354	443	540	651	740	886	1032	1196	1330	1530	1730	1947

Tabulka B.13: Metoda s nejmenší časovou náročností (s ARM DSP Library)

B.2 Metody predikce

B.2.1 Klasické operace vs. operace se submaticemi (bez ARM DSP Library)

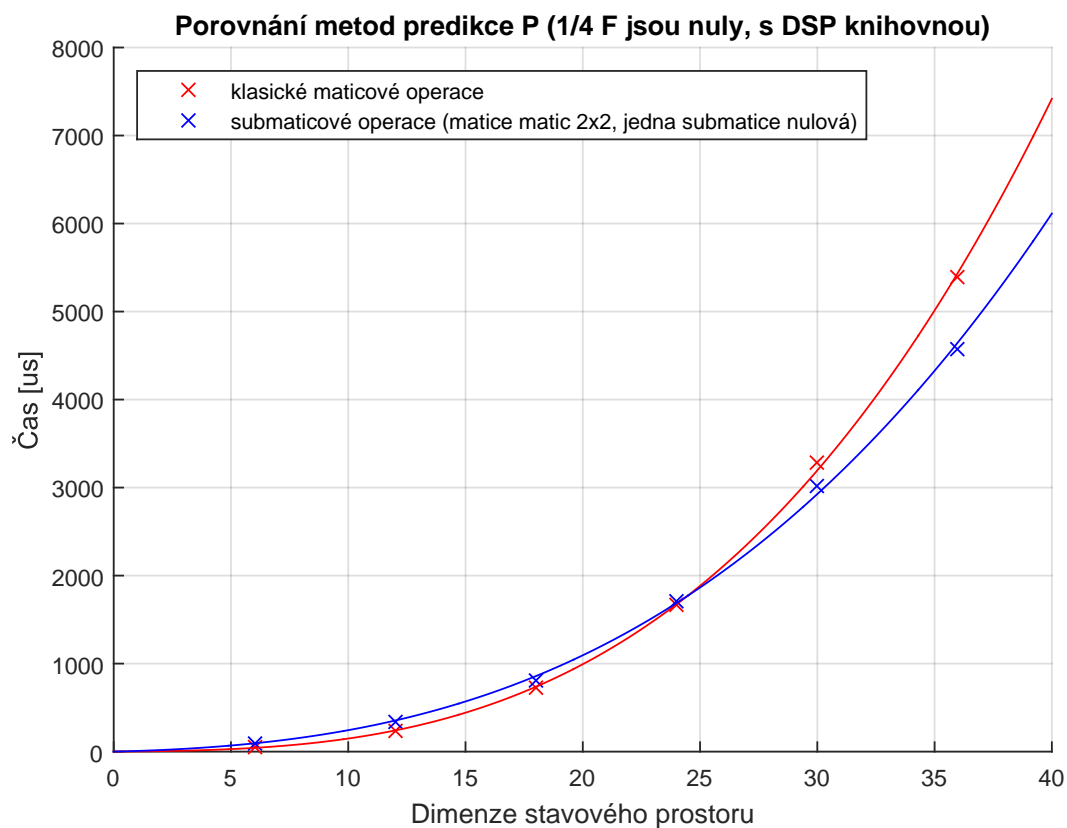


Obrázek B.3: Metody predikce (bez ARM DSP Library, 1/4 matice P jsou nuly)

Dimenze stav. prostoru	Operace se submaticemi [μ s]	Maticové operace [μ s]
6	453	333
12	2317	2300
18	6749	7400
24	14904	17135
30	27891	32994

Tabulka B.14: Metody predikce (bez ARM DSP Library, 1/4 matice P jsou nuly)

B.2.2 Klasické operace vs. operace se submaticemi (s ARM DSP Library)



Obrázek B.4: Metody predikce (s ARM DSP Library, 1/4 matice P jsou nuly)

Dimenze stav. prostoru	Operace se submaticemi [μs]	Maticové operace [μs]
6	102	46
12	340	237
18	815	719
24	1715	1659
30	3018	3278
36	4583	5395

Tabulka B.15: Metody predikce (s ARM DSP Library, 1/4 matice P jsou nuly)