Fakulta elektrotechnická
Katedra aplikované elektroniky a telekomunikací

Disertační práce

Určování polohy kolejových vozidel s využitím inerciální navigace a detekce charakteristických segmentů tratě

Autor: Ing. Lukáš Pušman
Školitel: doc. Ing. Jiří Skála, Ph.D.

Plzeň 2017

Abstrakt

Tato práce se zabývá návrhem systému pro určování polohy kolejových vozidel, založeném na inerciální navigaci a detekci charakteristických segmentů tratě. Popisovaný systém představuje alternativu k tradičnímu přístupu řešení navigační úlohy za pomoci Kálmánova filtru a pracuje primárně pouze s inerciálními daty. Hlavní myšlenka spočívá v porovnávání určitých charakteristických úseků tratě s předem známou trasou, která je u kolejových vozidel zpravidla daná. První část práce je zaměřena spíše teoreticky a jsou zde popsány základní principy, které využívají soudobé navigační systémy, problematika kolem satelitních navigačních systémů, inerciálních systémů, navigačních soustav a mapových podkladů. Dále je proveden rozbor metod vhodných právě pro navigaci kolejových vozidel a popis principu detekce charakteristických segmentů tratě. V této kapitole je také uvedeno několik odlišných přístupů, aplikovaných na drážní provoz. Další část textu je věnována popisu návrhu zařízení pro měření a sběr inerciálních dat a návrhu softwaru, který realizuje navrženou navigační metodu. Software je koncipován tak, aby pracoval s daty vzorek po vzorku tak, aby byla možná realizace systému v reálném čase. Poslední část textu pak popisuje provedená měření a pojednává o dosažených výsledcích a parametrech navrženého navigačního systému. Na základě těchto parametrů jsou zde uvedeny potenciální aplikace, pro které je navigační systém vhodný a zmíněny jsou také aplikace, ve kterých se o nasazení navrženého systému reálně uvažuje.

Klíčová slova

Navigační systém, určování polohy, kolejová vozidla, detekce segmentů tratě, globální družicový polohový systém (GNSS), protokol NMEA, inerciální měření, strap-down algoritmus, MEMS akcelerometr, MEMS gyroskop, polyfázový filtr, transformace souřadného systému, výpočet poloměru, software LabVIEW, veřejná doprava, tramvaj.

Abstract

Pušman, Lukáš. *Railway Vehicle Localization Based on Inertial Navigation and Specific Track Segment Detection* [*Určování polohy kolejových vozidel s využitím inerciální navigace a detekce charakteristických segmentů tratě*]. Pilsen, 2017. PhD thesis (in Czech). University of West Bohemia. Faculty of Electrical Engineering. Department of Applied Electronics and Telecommunications. Supervisor: Jiří Skála

---

In this thesis a design of a railway-vehicle localization system based on the inertial navigation and specific track-segment detection is described. The system offers an alternative to the traditional solution of the navigation task based on the Kalman filter and it uses primarily only the inertial data for its operation. The main idea is to detect and recognize some specific parts of the track and compare them with the whole track which is known in advance in case of railway. The first part of the thesis is focused on the theoretical background of the navigation and the satellite navigation systems, inertial systems, navigation frames and cartographic issues are described there. There are also described navigation methods which are suitable especially for the railway and the principle of specific track-segment detection in this part. Several different approaches of the railway navigation solution are introduced in this section as well. In the next part, design of a device for inertial measurement and data collection and design of a software which implements proposed navigation method are described. The software is designed to work with the data sample by sample and therefore a real-time application of the system is feasible. The last section of the thesis describes the performed measurements and results and achieved parameters of the navigation system. Based on the parameters potential applications for which the system is suitable are proposed and the applications for which an implementation of the designed system is actually considered are mentioned as well.

Keywords

Navigation system, localization, railway vehicle, track segment detection, Global Navigation Satellite System (GNSS), NMEA protocol, inertial measurement, strap-down algorithm, MEMS accelerometer, MEMS gyroscope, polyphase filter, navigation frame transformation, radius calculation, LabVIEW software, public transport, tram.

Prohlášení

Předkládám tímto k posouzení a obhajobě disertační práci, zpracovanou na závěr doktorského studia na Fakultě elektrotechnické Západočeské univerzity v Plzni.

Prohlašuji, že jsem svou závěrečnou práci vypracoval samostatně pod vedením vedoucího disertační práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené disertační práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících, autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 270 trestního zákona č. 40/2009 Sb.

Také prohlašuji, že veškerý software, použitý při řešení této disertační práce, je legální.

V Plzni dne 9. října 2017

Ing. Lukáš Pušman

. . . . . .

Podpis

Poděkování

Tato práce vznikla s podporou Ministerstva školství, mládeže a tělovýchovy ČR v rámci projektu RICE - Nové technologie a koncepce pro inteligentní systémy, číslo projektu LO1607 a grantů SGS-2012-019 (Moderní řešení elektronických řídících a informačních systémů) a SGS-2015-002 (Moderní metody řešení, návrh a aplikace elektronických a komunikačních systémů).

Obsah

Seznam obrázků

Seznam tabulek

Seznam symbolů a zkratek

ADC	Analog-to-Digital Converter. Analogově digitální převodník.
APP	Application Layer. Aplikační vrstva.
ARM	Acorn RISC Machine (Advanced RISC Machine). Typ architektury procesoru s redukovanou instrukční sadou.
ASCII	American Standard Code for Information Interchange. Americký standardní kód pro výměnu informací.
a_x	Dopředná složka zrychlení $\left[m{\cdot}s^{-2}\right]$.
a_y	Příčná složka zrychlení $\left[m{\cdot}s^{-2}\right]$.
a_z	Svislá složka zrychlení $\left[m{\cdot}s^{-2}\right]$.
BINR	Proprietární binární protokol společnosti NVS Technologies AG.
BPSK	Binary-Phase Shift Keying. Binární klíčování fázovým posuvem.
BSP	Board Support Package. Platformě závislá vrstva software.
BW	Bandwidth. Šířka pásma.
CAD	Computer-Aided Design. Počítačem podporovaný návrh.
CAN	Controller Area Network. Typ sériové komunikační sběrnice.
CDMA	Code Division Multiple Access. Kódový multiplex.
CLK	Clock. Hodinový signál.
COM	Communication port. Sériový port osobního počítače.
CPU	Central Processing Unit. Centrální procesorová jednotka.
CR	Carriage return. Netisknutelný řídící znak, který posune kurzor na začátek řádku.
CS	Chip Select. Signál pro výběr podřízeného obvodu.
CSV	Comma-Separated Values. Tabulka uložená jako textový soubor s hodnotami oddělenými čárkou.
CZEPOS	Síť permanentních stanic GNSS České Republiky.
ČÚZK	Český úřad zeměměřický a katastrální.
DFVLP	Data Flow Visual Programming Language. Grafický programovací jazyk respektující směr datového toku.
DGPS	Diferential GPS. Diferenciální GPS.
DLL	Dynamic-Link Library. Dynamicky linkovaná knihovna.
DLN	Data length. Délka dat.
DMA	Direct Memory Access. Přímý přístup do paměti.

DP	Filtr typu dolní propust.
DPS	Deska plošných spojů.
ECEF	Earth-Centered, Earth-Fixed. Kartézský souřadný systém se středem v hmotnostním těžišti Země.
ECS	Eddy Current Sensor. Senzor vířivých proudů.
EGNOS	European Geostationary Navigation Overlay Service. Evropská síť pozemních stanic GNSS.
ENU	East North Up. Orientace pravotočivé kartézské navigační soustavy ve směrech východ, sever a vzhůru.
EPSG	European Petroleum Survey Group. Vědecká organizace s vazbou k evropskému naftovému průmyslu (1986 - 2005).
FAA	Federal Aviation Administration. Federální letecký úřad v USA.
FAT	File Allocation Table. Typ souborového systému.
FDMA	Frequency Division Multiple Access. Frekvenční multiplex.
FIFO	First in, first out. První dovnítř, první ven (metoda pro práci s daty).
FIR	Finite Impulse Response. Filtr s konečnou impulzní odezvou.
FPGA	Field-Programmable Gate Array. Programovatelné hradlové pole.
GCC	GNU Compiler Collection. Sada překladačů vytvořených v rámci projektu GNU.
GGA	Global Positioning System Fix Data. Rozšířená sada údajů, kterou poskytují GNSS přijímače.
GLONASS	Globalnaja navigacionnaja sputnikovaja sistěma. Globální navigační družicový systém (SSSR/Rusko).
GNSS	Global Navigation Satellite System. Globální družicový polohový systém.
GNU	GNU's Not Unix. Projekt zaměřený na svobodný software, inspirovaný operačními systémy unixového typu.
GPIB	General Purpose Interface Bus. Komunikační rozhraní pro měřicí přístroje dle standardu IEEE-488.
GPIO	General-Purpose Input/Output. Obecná vstupně/výstupní brána.
GPL	General Public License. Licence pro svobodný software vytvořená v rámci projektu GNU.
GPS	Global Positioning System. Globální polohový systém (USA).
GUI	Graphical User Interface. Grafické uživatelské rozhraní.
HAL	Hardware Abstraction Layer. Hardwarová abstraktní vrstva.
HP	Filtr typu horní propust.
CHS	Checksum. Kontrolní součet.

I/O	Input/Output. Vstupně-výstupní.
I2C	Inter-Integrated Circuit. Typ sériové komunikační sběrnice.
IEC	International Electrotechnical Commission. Mezinárodní elektrotechnická komise.
IMU	Inertial Measurement Unit. Jednotka pro měření inerciálních veličin.
INT	Interrupt. Přerušení.
IOGP	International Association of Oil & Gas Producers. Mezinárodní asociace producentů ropy a zemního plynu (nástupce EPSG).
IRM	International Reference Meridian. Mezinárodní referenční poledník.
IRP	International Reference Pole, Mezinárodní referenční pól.
ISO/OSI	International Organization for Standardization/Open Systems Interconnection. Referenční model ISO/OSI.
J-TAG	Joint Test Action Group. Architektura Boundary-Scan dle standardu IEEE 1149.1.
LabVIEW	Laboratory Virtual Instrument Engineering Workbench. Vývojové prostředí od společnosti National Instruments.
LF	Line feed. Netisknutelný řídicí znak, který posune kurzor na další řádek.
LGA	Land Grid Array. Typ pouzdra integrovaného obvodu pro povrchovou montáž.
LQFP	Low profile Quad Flat Package. Typ pouzdra integrovaného obvodu pro povrchovou montáž.
MAV	Moving average. Plovoucí průměr.
MCU	Microcontroller unit. Jednočipový mikropočítač.
MCX	Micro Coaxial. Typ koaxiálního konektoru.
MEMS	Micro Electro Mechanical Systems. Mikroelektromechanické systémy.
MSB	Most Significant Bit. Nejvýznamnější (nejvyšší) bit.
MWR	Middleware. Rozhraní mezi aplikační a platformě závislou vrstvou.
NED	North East Down. Orientace pravotočivé kartézské navigační soustavy ve směrech sever, východ a dolů.
NI	National Instruments. Společnost vyrábějící zařízení a software pro automatizované testování.
NMEA	National Marine Electronics Association. Národní sdružení pro lodní elektroniku.
ODR	Output Data Rate. Frekvence výstupních dat.

OEM	Original Equipment Manufacturer. Výrobce, jehož produkt je prodáván pod jinou obchodní značkou.
OSM	OpenStreetMap. Projekt pro tvorbu volně dostupných geografických dat a map.
PMDP	Plzeňské městské dopravní podniky a.s.
R/W	Read/Write. Zápis a čtení.
RAM	Random-Access Memory. Paměť s přímým přístupem (operační paměť).
RDS	Radio Data System. Systém pro přenos doplňkových informací v sítích VKV FM.
RLG	Ring Laser Gyroscope. Konstrukce gyroskopického senzoru na optickém principu.
RMC	Recommended Minimum Navigation Information. Základní sada údajů, kterou poskytují GNSS přijímače.
RSSI	Received Signal Strength Indication. Indikace síly přijímaného signálu.
RTCM	Radio Technical Commission for Maritime Services. Mezinárodní organizace zabývající se standardizací v oblasti telekomunikace (USA).
RTOS	Real-Time Operating System. Operační systém reálného času.
S-JTSK	Systém jednotné trigonometrické sítě katastrální.
SBUS	Satellite Based Augmentation System. Systém pro zvýšení přesnosti GNSS distribuující navigační data pomocí družic.
SD	Secure Digital. Typ paměťové karty.
SDIO	Secure Digital Input Output. Periferie pro komunikaci s kartami SD (MCU STMicroelectronics).
SMA	SubMiniature version A. Typ koaxiálního konektoru.
SOF	Start Of Frame. Znak charakterizující začátek rámce.
SPI	Serial Peripheral Interface. Typ sériové komunikační sběrnice.
SPL	Standard Peripheral Libraries. Knihovna pro práci s periferiemi MCU STM32 (STMicroelectronics).
SRAM	Static Random-Access Memory. Statická paměť RAM.
ST	STMicroelectronics. Výrobce elektroniky a polovodičových součástek (Švýcarsko).
TTL	Transistor-Transistor Logic. Tranzistorově-tranzistorová logika.
UART	Universal Asynchronous Receiver/Transmitter. Univerzální asynchronní komunikační rozhraní.
UAV	Unmanned Aerial Vehicle. Bezpilotní letadlo.
USB	Universal Serial Bus. Univerzální sériové rozhraní.
UTC	Coordinated Universal Time. Koordinovaný světový čas.

VI	Virtual Instrument. Označení a přípona souboru programu vytvořeného v prostředí LabVIEW.
VISA	Virtual Instrument Software Architecture. Rozhraní pro komunikaci mezi PC a měřicími přístroji podporující různé protokoly a sběrnice.
VPL	Visual Programming Language. Grafický programovací jazyk.
WAAS	Wide Area Augmentation System. Systém pro zpřesnění GPS pro letectví distribuující navigační data pomocí družic.
WGS84	World Geodetic System 1984. Světový geodetický systém 1984.
φ	Úhel klonění (roll) $[rad]$, případně $[°]$.
θ	Úhel klopení (pitch) $[rad]$, případně $[°]$.
ψ	Úhel bočení (yaw) $[rad]$, případně $[°]$.
ω_φ	Úhlová rychlost klonění $\left[rad{\cdot}s^{-1}\right]$, případně $\left[°{\cdot}s^{-1}\right]$.
ω_θ	Úhlová rychlost klopení $\left[rad{\cdot}s^{-1}\right]$, případně $\left[°{\cdot}s^{-1}\right]$.
ω_ψ	Úhlová rychlost bočení $\left[rad{\cdot}s^{-1}\right]$, případně $\left[°{\cdot}s^{-1}\right]$.

1

Úvod

Předmětem této práce je návrh navigačního systému založeného na principu identifikace charakteristických segmentů tratě, který primárně využívá pouze data získaná měřením inerciálních veličin. Systém je zaměřen především na určování polohy kolejových vozidel, protože je zde možné definovat určité úseky - segmenty a popsat je pomocí parametrů. Databáze těchto segmentů je pak jedním ze vstupů navigačního algoritmu. Pro měření inerciálních veličin využívá navržené zařízení běžné integrované senzory, ale také obsahuje přijímač globálního družicového polohového systému (GNSS) pro alternativní funkci, kdy jsou kombinována jak inerciální tak i GNSS data. Popisovaný navigační systém je určen pro dispečerské a informační systémy v rámci městské hromadné dopravy či jiných menších dopravních sítí a nemá ambice pro nasazení v bezpečnostně relevantních aplikacích.

Ačkoli by se mohlo zdát, že navigační úloha je v dnešní době již zcela vyřešená problematika a osobní navigace je implementována i v tak běžných zařízeních jako jsou mobilní telefony či náramkové hodinky, je to na úrovni veřejné dopravy a průmyslových aplikací stále aktuální a celosvětově řešené téma a jak naznačuje současná situace, bude nasazení GNSS systémů v rámci železniční dopravy pravděpodobně ještě dlouhou dobu diskutováno a vyvíjeno.

Jak již bylo zmíněno, navrhovaný navigační systém není určen pro bezpečnostní aplikace v rámci železnice, ale představuje řešení pro monitorovací a dispečerské systémy v rámci menších dopravních sítí. Z navigačního systému je zde možné získat další provozní údaje jako je například doba stání v zastávce, časy průjezdů taktovacími body, plynulost a styl jízdy a z těchto dat vytvářet statistiky. Výsledky pak lze například využít pro zlepšení plánování přepravy a získat tak i ekonomický přínos. Takovéto aplikace mohou tedy být dalším důvodem pro vývoj popisovaného navigačního systému.

Dalším nezanedbatelným důvodem, proč má tato práce smysl, je možnost spojení navigačních dat s provozními daty vozidla, jako je například proudový odběr z napájecí sítě nebo baterií, údaje o spotřebovávaném či rekuperovaném výkonu apod. Tato data lze tedy spojit s časem a polohou, resp. projížděným terénem a získat tak statistické údaje využitelné v rámci návrhu vozidla jeho pohonu, dimenzování baterií a kondenzátorových polí, napájecí sítě atd. Také je možné tato data využít v rámci oblasti elektromobility,

která je dnes velmi aktuální problematikou.

Myšlenka použít běžné levné senzory a pomocí kombinace matematických výpočtů a souboru určitých podmínek z jejich výstupu získat relevantní informace, byla asi hlavní motivací po celou dobu práce. Hlavním cílem je tedy realizace funkčního zařízení, které dokáže vykonávat navigační úlohu dlouhodobě, bez příjmu GNSS signálu. Dále je cílem provést sérii měření v reálném provozu a zhodnocení výsledků tak, aby bylo možné stanovit dosažené parametry navrženého navigačního systému. Celkově by se cíle práce daly shrnout v následujících bodech:

- Prostudování problematiky určování polohy (inerciální a GNSS systémy, souřadné systémy a mapové podklady)
- Výběr a implementace metod vhodných pro kolejová vozidla
- Návrh a realizace zařízení pro měření a sběr dat
- Implementace navržených navigačních algoritmů v rámci softwaru pro zpracování dat
- Provedení měření v reálném provozu a vyhodnocení dosažených parametrů navrženého navigačního systému

Protože systém může využívat také navigační data z GNSS, jsou v první části textu stručně popsány GNSS systémy a protokol NMEA, který slouží pro komunikaci s přijímači. Dále je zde vysvětlen pojem inerciální měření a podrobně rozebrána problematika kolem senzorů, zejména s ohledem na jejich přesnost a možné chyby. Jsou zde uvedeny základní vztahy pro výpočet dalších veličin, včetně výpočtu použitého pro realizaci tzv. strapdown systému. Také je zde popsán návrh číslicového filtru použitého v navrženém zařízení. V závěru této převážně teoretické části textu je také zmíněna problematika kolem formátu navigačních dat z GNSS, souřadných systémů, projekcí na mapové podklady a otázka kolem mapových podkladů samotných. Matematické výpočty a transformace, které jsou zde uvedeny, jsou také implementovány v popisovaném systému.

Druhá část textu se již zabývá vlastní problematikou určování polohy kolejových vozidel a jsou zde popsány principy, které lze s výhodou uplatnit tak, aby mohl systém pracovat dlouhodobě i bez příjmu GNSS dat. Je zde popsán princip detekce charakteristických segmentů a uveden matematický aparát použitý v hlavních algoritmech navigačního programu, jako je transformace souřadného systému do soustavy orientované vůči Zemi a vůči směru jízdy vozidla. V závěru této části jsou také uvedeny některé další publikované práce, které pro navigaci kolejových vozidel využívají odlišné principy a přístupy.

Následující dvě kapitoly textu jsou celkově rozsáhlejší a popisují návrh a realizaci zařízení pro měření a sběr dat, které představuje převážně hardwarovou část popisovaného systému a dále zpracování naměřených dat, které reprezentuje softwarovou část práce a implementaci matematických výpočtů nad změřenými daty. Popis měřicího zařízení

zahrnuje návrh prvního i druhého prototypu, parametry zvolených senzorů a modulu GNSS přijímače, strukturu a schéma firmwaru pro MCU a popis mechanického provedení. Část zabývající se popisem softwaru nejprve stručně uvádí do prostředí LabVIEW ve kterém byly všechny programy vytvořeny a také krátkým popisem pomocných aplikací vytvořených pro první analýzy naměřených dat. Další část textu pak detailně popisuje hlavní navigační program a jeho podprogramy, které v sobě implementují matematické výpočty a algoritmy popsané v předchozích kapitolách.

V poslední části tohoto textu jsou popsána měření, která byla s navrženým navigačním systémem postupně prováděna, a také výsledky a výstupy, které je možné ze systému získat. Výsledky všech měření, resp. jim odpovídající výstupy systému, byly podrobně analyzovány za účelem získání statistických dat a vyhodnocením dosažených parametrů systému. Vlastní vyhodnocení, postihující zejména dosaženou přesnost nebo chybovost navigačního systému, je rovněž prezentováno v této kapitole v podobě grafů a konkrétních hodnot.

2

Určování polohy na zemském povrchu

Určování polohy a navigace na Zemi provází lidstvo v různých podobách prakticky od jeho počátků. V dnešní době je díky technologiím tento obor přístupný široké veřejnosti a zdaleka nevyžaduje takové znalosti a přístroje jako tomu bylo v minulosti. Díky systémům GNSS, kompaktním přenosným zařízením a serverům s mapovými podklady pro celý zemský povrch, může být určování polohy považováno za samozřejmost.

Mimo již zmíněné satelitní navigační systémy, resp. GNSS přijímače, obsahuje většina navigačních aplikací také čidla pro inerciální měření a s tím spojený matematický aparát. Díky tomu je možné data z GNSS dále zpřesnit, zvýšit jejich hustotu nebo získat další užitečné informace. Také je třeba vhodným způsobem zpracovat výstupní signály senzorů a potlačit tak rušení a chyby, které jsou v nich obsaženy. S touto problematikou souvisí především návrh vhodného filtru, s ohledem na dnešní hardware patrně číslicového.

Další součástí navigační úlohy je v mnoha aplikacích projekce údajů o poloze do některého z mapových podkladů. Tato operace vyžaduje přepočet souřadnic mezi jednotlivými souřadnými soustavami, tak aby standardizovaný formát dat z GNSS přijímače bylo možné zobrazit na mapě v dané projekci.

Ačkoli jsou uvedené principy značně obecné, platí i pro konkrétní příklad navigace, který je řešen v rámci tohoto textu a je nutné je detailněji popsat. Právě popis výše uvedených systémů a postupů, s ohledem na aplikaci určování polohy kolejových vozidel, je předmětem této kapitoly.

2.1 Systémy GNSS

Globální družicové polohové systémy již dnes představují standardní způsob určování polohy na Zemském povrchu. Masivní nástup této technologie započal uvolněním přesnější

verze GPS pro civilní využití v roce 2000[1]. To umožnilo používat GPS navigaci pro osobní účely v automobilech, turistice a později, při společném použití GPS a konkurenčního GLONASS i například v civilním letectví. Z hlediska popisované aplikace nás zajímá především přesnost těchto systémů a podmínky, za kterých je tato přesnost dosažena.

2.1.1 Global Positioning System

Global Positioning System (GPS) představuje vojenský navigační systém budovaný od roku 1973 v USA, který dosáhl celosvětového pokrytí začátkem 90. let. Následně se také ukázal možný potenciál celého systému a bylo schváleno použití GPS i pro civilní účely. Jak již bylo zmíněno výše, od poloviny roku 2000 pak byla uvolněna přesnější verze systému, takže se přesnost pro civilní aplikace ještě asi desetinásobně zvýšila. [2]

Kosmický segment systému GPS v současnosti tvoří 24 družic, z nichž tři jsou provozovány jako záložní. Družice se pohybují po šesti oběžných drahách skloněných po 60° ve výšce asi 20000 km. K určení polohy je třeba zachytit signál minimálně ze tří, resp. čtyř družic, přičemž s rostoucím počtem viditelných družic roste i přesnost určení polohy. [1]

Do ukončení záměrného rušení GPS dat byla přesnost určení horizontální polohy pro civilní potřeby přibližně 100 m. Nyní jsou i nejlevnější přístroje schopné určit polohu s přesností přibližně 10 m. V praxi to znamená, že při dobrém nebo jen mírně zhoršeném příjmu se přesnost pohybuje od 10 m do asi 20 m a při výrazně zhoršeném příjmu se může zhoršit i na 90 m nebo může dojít k úplnému výpadku. Z hlediska vertikální polohy (určení nadmořské výšky) platí, že přesnost je ještě asi o 50% horší než u polohy vertikální. Z tohoto důvodu bývají komerční přístroje vybavené například ještě tlakovým výškoměrem apod. Jinou metodou zpřesňování GPS je tzv. diferenciální GPS (DGPS), kde se využívá síť pevných pozemních stanic, které mají dlouhodobě zjištěnou přesnou polohu. Tyto stanice generují korekční data na základě porovnání své polohy s přijatými GPS daty a distribuují korekční informace mobilním zařízením ve svém okolí. Pro distribuci může sloužit mnoho kanálů, například síť internet, datové přenosy v mobilních sítích, dlouhovlnné vysílání, RDS apod. Jako příklad může být uvedena česká síť 27 stanic zvaná CZEPOS, která je provozována zeměměřičským úřadem a která umožňuje provádět statická geodetická měření s přesností jednotek centimetrů. [3]

2.1.2 Globalnaja navigacionnaja sputnikovaja sistěma

Vývoj systému Globalnaja navigacionnaja sputnikovaja sistěma (GLONASS) započal v SSSR zhruba ve stejné době jako vývoj GPS a jeho vojenský účel byl shodný. Na rozdíl od GPS ale nebyl dokončen v plném celosvětovém měřítku a vývoj stagnoval až do roku 2001. Od roku 2001 začala obnova systému, takže v roce 2011 bylo dosaženo celosvětového pokrytí a od roku 2012 je systém prakticky využíván komerčními zařízeními. [4]

[1]Systém GPS až do 1.5.2000 záměrně poskytoval informace o poloze se sníženou přesností a plná přesnost byla dostupná pouze pro vojenské účely. [1]

Kosmický segment GLONASS je složen z 24 družic, které obíhají po třech oběžných drahách ve výšce 19100 km. Dráhy mají sklon 65° a jsou vzájemně posunuty o 120°. Každá dráha je rozdělena po 45° na 8 pozic, v nichž se nachází družice. Kosmický segment GLONASS se od konkurenční GPS liší nejen uspořádáním družic, ale také způsobem vysílání a modulace rádiového přenosu[2]. Pro určení polohy je třeba přijímat signál z pěti družic systému, přičemž přesnost civilní verze se pohybuje kolem 50 m. [4]

2.1.3 Ostatní systémy

Sem lze zařadit zejména evropský systém Galileo, který je momentálně ve výstavbě a který by měl plné funkce dosáhnout na přelomu let 2019 a 2020. Kosmický segment je plánován na 30 družic a základní přesnost je 4 m. Mimo volně přístupné verze systému se základní přesností je plánována také zpoplatněná, dosahující přesnosti lepší než jeden metr. Systém Galileo také využívá pro zpřesnění polohy stávající družice GPS a GLONASS. [5]

Za zmínku stojí ještě čínský navigační systém Beidou, který je v provozu od roku 2008 a využívá geostacionární družice. Pokrytí systému Beidou 1 zahrnuje pouze Čínu a civilní přesnost je 10 m. Čína dále plánuje výstavbu druhé verze systému Beidou 2, která by měla být dokončena kolem roku 2020 a která bude se svými 30 satelity pokrývat již celou Zemi. Přesnost tohoto systému pro civilní sektor by měla být do 10 m. [6]

2.1.4 Protokol NMEA

Pro výše popsané GNSS systémy jsou na trhu dostupné různé přijímače, ať už jako OEM moduly určené k další zástavbě nebo jako zařízení pro koncového zákazníka. Vlastní přijímač pak obsahuje nejen rádiovou vysokofrekvenční část přijímající signál z družic, ale také mikropočítač, který provádí navigační výpočty a další funkce. Protože existuje více výrobců přijímačů a celá řada zařízení do kterých jsou tyto přijímače integrovány, byl standardizován komunikační protokol NMEA, s jehož pomocí lze spojit přijímač s dalším systémem, jako je např. CPU mobilního zařízení nebo obecně jiný nadřazený počítač.

Protokol NMEA 0183 (IEC 61162-1) je standardním rozhraním, které je využíváno k předávání informací o poloze, rychlosti, kvalitě příjmu i počtu viditelných satelitů a většina komerčně dostupných přijímačů jej podporuje. Základem protokolu jsou tzv. věty (sentences), které se v zásadě dělí na vstupní a výstupní. Vstupní slouží k nastavování přijímače a nejsou tak rozšířené (podporují je jen některá zařízení). Výstupní pak lze rozdělit do celé řady skupin a obsahují poziční a navigační data, informaci o aktuální konstelaci satelitů, stavu přijímače apod. Věty jsou posílány v ASCII režimu s definovaným počátečním znakem a jsou ukončeny odřádkováním[3]. Toho lze s výhodou využít při

[2]Systém GLONASS používá frekvenční modulaci FDMA, která je náročnější na konstrukci přijímače i šířku frekvenčního pásma, na rozdíl od GPS, kde je využívána fázová modulace BPSK a CDMA multiplex. [4]

[3]Odřádkováním jsou zde myšleny dva znaky, tedy Carriage Return (CR) a Line Feed (LF).

čtení sériové linky, kdy odřádkování může být použito jako delimiter. Další zpracování přijaté věty ve formě textového řetězce je založené na principu tzv. tokenů, které jsou odděleny středníkem. Na konci každé věty je ještě odesílán kontrolní součet (checksum) pro případnou validaci přijatých dat. Vzhledem k velkému množství vět, které jsou v rámci NMEA 0183 definovány, nemá smysl provádět zde jejich výčet a popis. V případě zájmu jsou tyto informace dostupné například v [7].

Z hlediska aplikace, kterou se zabývá tato práce, je zajímavá věta Recommended Minimum Navigation Information (RMC), která je asi nejrozšířenější a podporují ji všechna zařízení. Struktura RMC je zachycena na obrázku 2.1 a význam jednotlivých polí je následující: 1 - časový údaj v UTC, 2 - A nebo V (A znamená že GNSS data jsou platná), 3 - zeměpisná šířka, 4 - N nebo S (severní nebo jižní), 5 - zeměpisná délka, 6 - E nebo W (východní nebo západní), 7 - rychlost vůči zemskému povrchu v uzlech, 8 - zeměpisný kurs, 9 - datum ve formátu ddmmyy, 10 - magnetická deklinace, 11 - E nebo W (východní nebo západní), 12 - FAA režim[4], 13 - Kontrolní součet (checksum). První dvě písmena za počátečním znakem označují název družicového systému, ze kterého data pocházejí. Na obrázku jsou to znaky GP, které označují systém GPS a dále to mohou být znaky GL pro GLONASS, GA pro Galileo a další.

```
        1          2 3        4 5         6  7   8   9    10 11 12  13
        |          | |        | |         |  |   |   |    |  | |   |
$GPRMC,hhmmss.ss,A,llll.ll,a,yyyyy.yy,a,x.x,x.x,xxxx,x.x,a,m,*hh<CR><LF>
```

Obr. 2.1: Struktura NMEA věty RMC [7]

Další NMEA věta, která je implementována ve většině GNSS přijímačů je Global Positioning System Fix Data (GGA). Tato věta obsahuje, mimo jiné, také informaci o nadmořské výšce, která je vztažená ke geoidu[5] a rozdílovou hodnotu mezi touto výškou a výškou vztaženou k referenčnímu elipsoidu WGS84 [9]. Struktura GGA je zachycena na obrázku 2.2 a význam jednotlivých polí je následující: 1 - časový údaj v UTC, 2 - zeměpisná šířka, 3 - N nebo S (severní nebo jižní), 4 - zeměpisná délka, 5 - E nebo W (východní nebo západní), 6 - kvalita příjmu (hodnota 0 až 8, podle kvality příjmu a režimu přijímače), 7 - Počet viditelných satelitů (hodnota 0 až 12), 8 - přesnost v horizontálním směru (v metrech), 9 - nadmořská výška (vztaženo ke geoidu), 10 - jednotky výšky (metry), 11 - rozdíl mezi udávanou výškou a elipsoidem WGS84 (záporná hodnota znamená že výška je pod povrchem elisoidu), 12 - jednotky rozdílu (metry), 13 - stáří DGPS dat (v sekundách), 14 - ID stanice DGPS, 15 - Kontrolní součet (checksum).

[4]FAA režim byl přidán od NMEA verze 2.3 a má následující hodnoty: A - Autonomous mode, D - Differential Mode, E - Estimated (dead-reckoning) mode, M - Manual Input Mode, S - Simulated Mode, N - Data Not Valid a P - Precise. [7]

[5]Geoid je fyzikální model povrchu Země při střední hladině světových oceánů. [8]

```
        1         2       3 4        5 6 7  8   9  10 11 12 13  14   15
        |         |       | |        | | |  |   |   | |  | |  |    |
$GPGGA,hhmmss.ss,llll.ll,a,yyyyy.yy,a,x,xx,x.x,x.x,M,x.x,M,x.x,xxxx*hh<CR><LF>
```

Obr. 2.2: Struktura NMEA věty GGA [7]

2.2 Inerciální měření

Pojem inerciální měření v tomto kontextu představuje měření zrychlení a úhlových rychlostí v pravoúhlé kartézské soustavě pevně spojené s navigovaným tělesem. Slovní spojení pochází z anglického termínu Inertial Measurement Unit (IMU), který představuje zařízení vykonávající takovéto měření.

Původně byla jednotka IMU součástí čistě inerciálních navigačních systémů, které byly intenzivně vyvíjeny od 40. let minulého století a využívaly rotační gyroskopy a stabilizaci pohyblivé plošiny se senzory pomocí serv (gimballed inertial platform). Později, s možností použití elektroniky s větším výpočetním výkonem, začaly inerciální navigační systémy využívat obrácený princip, kdy jsou senzory umístěny na pevné podložce[6] a hodnoty z gyroskopů jsou použity k výpočtu směrového vektoru a transformační matice pro hodnoty zrychlení. Toto provedení IMU výrazně zjednodušuje mechanickou konstrukci původního provedení s naklápěnou plošinou, ale také klade vyšší nároky na použité senzory, zejména na dynamický rozsah, přesnost a linearitu gyroskopů. [10]

Pro ilustraci jsou oba systémy IMU blokově znázorněny na obrázku 2.3. Funkce obou typů IMU již byla popsána výše, ale za zmínku stojí ještě nutnost korekcí různých chyb, které v systému vznikají, jak je to uvedeno na schématu 2.3 a). Další korekcí je tzv. Schulerova kompenzace [11], která zohledňuje kulový tvar Země. Důsledkem této kompenzace je naklápění plošiny tak, aby směr osy akcelerometru nahoru-dolů byl shodný se směrem tíhového zrychlení. Toto je implementováno výpočetně i v systému strap-down, viz. obrázek 2.3 b). Mezi oběma systémy je ještě patrný rozdíl v použitých gyroskopech. Zatímco ve starších systémech s naklápěnou plošinou převládalo použití klasických rotačních gyroskopů, novější strap-down systémy jsou ve většině případů založeny na gyroskopech pracujících na jiném principu, např. optické vlákno [12], Ring Laser Gyroscope (RLG) [10] nebo vibrující element (MEMS) [12]. Tyto gyroskopy ze svého principu nevrací přímo hodnotu úhlů klonění φ, klopení θ a bočení ψ, ale hodnoty úměrné úhlové rychlosti ω_φ, ω_θ a ω_ψ. Tyto gyroskopické senzory bývají označovány jako tzv. rate-gyro.

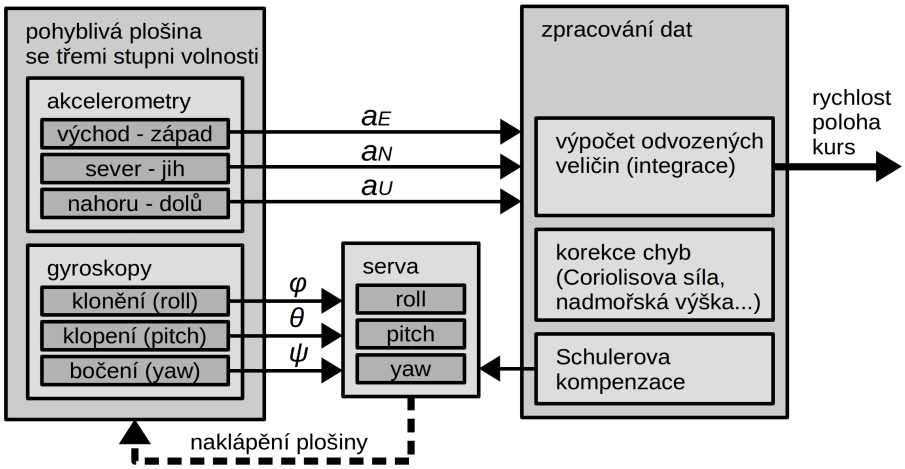
V dnešní době se výše popsané, čistě inerciální navigační systémy většinou kombinují se systémy GNSS[7], což zajišťuje pravidelnou verifikaci vypočítané polohy vůči zemskému souřadnému systému a akumulativní chyba inerciální navigace je tak prakticky vyloučena[8]. Kombinace IMU a GNSS také umožnila v některých aplikacích nasazení méně

[6] Senzory jsou fixovány vůči navigovanému tělesu.

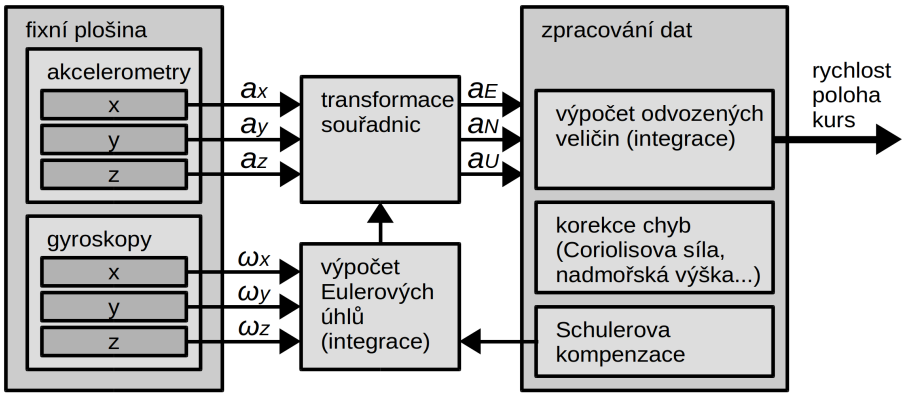
[7] Jen tam, kde to má smysl. Podzemní či podmořské aplikace toto stále vylučují.

[8] Pro srovnání, špičkové, čistě inerciální systémy mají akumulativní chybu v určení polohy menší než 1 km za hodinu. [10]

přesných senzorů, které jsou výrazně levnější a snadněji integrovatelné. Díky tomu, disponují dnes určitou formou navigačního systému i taková zařízení, jako jsou mobilní telefony, automobilové navigace nebo náramkové hodinky. Zde jsou nasazeny levné integrované senzory vyráběné masově technologií MEMS, ale ve spojení s GNSS je jejich výkon dostatečný. Toto ale zdaleka neplatí pro všechny aplikace a například v letectví se stále používají IMU s rotačními gyroskopy, případně s optickými gyroskopy technologie RLG. [10]



(a) IMU s pohyblivou plošinou



(b) IMU typu strap-down

Obr. 2.3: Bloková schémata obou známých provedení IMU [10]

2.2.1 Senzory

Senzory používané v popisované, polohu-určující aplikaci jsou akcelerometry a gyroskopy vyrobené technologií MEMS a jde tedy o integrované obvody. Jejich konkrétní výběr a parametry budou uvedeny později, během popisu návrhu hardwarové části projektu.

2.2.1.1 MEMS akcelerometry

Akcelerometry MEMS jsou založené na principu detekce vychýlení seismického tělíska pružně uloženého v pouzdře senzoru, který je pevně spojen s pohybujícím se objektem. [12]

Takovou strukturu je možné technologií MEMS vyrobit i v křemíkovém waferu a tedy integrovat. Nejčastější provedení struktury je takové, že seismické tělísko destičkovitého tvaru je pružně uloženo mezi dalšími dvěma destičkami, navzájem elektricky izolovanými. Tato struktura pak tvoří kondenzátor, případně dva sériově zapojené kondenzátory, jejichž kapacita je navzájem spřažená a proměnlivá v závislosti na vychýlení střední elektrody (tělíska) k jedné nebo druhé krajní elektrodě. Změna kapacity je pak úměrná zrychlení, které působí na tělísko, tedy i na celý senzor.

Protože výchylka tělíska dosažitelná v křemíkové struktuře je řádově 10 až 20 μm [12], je měření zrychlení velmi náchylné na veškeré nelinearity, chyby a rušení, které se mohou v systému vyskytovat. Tento problém bývá řešen použitím ještě jednoho identického kondenzátoru vytvořeného ve stejné struktuře. Výstupy obou kondenzátorů pak lze zpracovávat diferenciálně, zapojené v opačném smyslu. Zrychlení je pak úměrné rozdílu kapacit obou kondenzátorů a prvotní nelinearita je tak kompenzována. Tento princip také částečně řeší teplotní kompenzaci senzoru.

Další možností, kterou MEMS akcelerometry nabízí je automatická kalibrace, kterou lze provést tak, že pomocné obvody senzoru připojí mezi elektrody kondenzátoru napětí a tím elektrostaticky vychýlí tělísko do definované polohy. Hodnota na výstupu akcelerometru by pak měla odpovídat hodnotě dané výrobcem a pokud ne, lze provést kompenzaci rozdílovou hodnotou během zpracování signálu. [12]

2.2.1.2 MEMS gyroskopy

Gyroskopy MEMS jsou ze své podstaty vždy typu rate-gyro a jejich výstup je tedy úměrný úhlové rychlosti, kterou je senzorem otáčeno. Toto je dáno tím, že MEMS technologií nelze integrovat rotační element jaký nalezneme u klasických mechanických gyroskopů, ale místo něj je integrován vibrující element. Mechanicky jde tedy o podobné řešení jako u MEMS akcelerometrů. Systém využívá působení Coriolisovy síly [13], respektive zrychlení, které tato síla vyvolává. Vibrující tělísko se během oscilací pohybuje lineárně ve vztažné soustavě senzoru, kterou je ale, vlivem vnější síly, otáčeno. Toto otáčení pak vyvolá Coriolisovo zrychlení, které působí kolmo na lineární pohyb tělíska a je úměrné jeho rychlosti a rychlosti otáčení senzoru. [12]

Na rozdíl od akcelerometru, existuje v případě gyroskopu několik odlišných konstrukčních provedení, které se všechny vyrábí v závislosti na výrobci a na určení obvodu. Oscilační pohyb tělíska pružně upevněného k podpůrnému rámečku je realizován elektrostatickým vychylováním podobně, jako je tomu u výše popsaných akcelerometrů během kalibrace. Podpůrný rámeček je opět pružně uložen v tělese senzoru a začne-li se senzor otáčet, na rámeček začne působit Corriolisova síla. Rámeček je pak rozkmitán v kolmém směru k budícím kmitům tělíska, se shodnou frekvencí a s amplitudou úměrnou úhlové rychlosti otáčení. Mechanické kmitání je převáděno na elektrický signál stejně jako u akcelerometru, tedy na principu měnící se kapacity. [12]

2.2.1.3 Výběr senzorů a jejich uspořádání

Integrované obvody MEMS akcelerometrů a gyroskopů jsou vyráběny buď ve výše popsané formě, tedy s rozdílovým zesilovačem a analogovým napěťovým výstupem, případně ještě s obvody pro kalibraci, režim spánku apod. nebo jsou integrovány do složitějších číslicových obvodů, kde je realizováno vzorkování a filtrování signálů z vlastních senzorů a číslicově zpracovaná data jsou pak dostupná pomocí jedné ze standardních komunikačních sběrnic typu SPI, I2C apod. O vhodnosti nasazení analogového nebo číslicového obvodu pak rozhoduje zejména požadavek na vzorkovací kmitočet měření resp. kmitočet výstupních dat (ODR) u digitálních senzorů a také to, zda je dostupná dostatečná dokumentace k filtrům použitým uvnitř obvodu a ke způsobu jejich konfigurování. Například senzory vyráběné pro mobilní elektroniku, které jsou zpravidla digitální, mají vnitřní zpracování signálu uzpůsobeno především k detekování volného pádu, poklepání na displej apod., což nemusí v jiných aplikacích vyhovět.

Prvním senzorem použitým v navržené IMU je digitální MEMS akcelerometr, jehož výstupem jsou tři číselné hodnoty, které jsou lineárně[9] úměrné velikosti zrychlení, respektive třem jeho složkám, které působí v definovaných směrech x, y a z vůči čipu senzoru. Ze znalosti obvodu a z informací, které poskytuje výrobce je pak možné tato čísla převést na skutečné hodnoty ve správném měřítku. Máme tedy k dispozici hodnoty zrychlení a_x, a_y a a_z a při vhodné orientaci senzoru bude platit, že složka x reprezentuje silové působení v podélném směru (ve směru jízdy vozidla), složka y v příčném směru (kolmo na směr jízdy) a složka z ve svislém směru. Informace o zrychlování a brzdění kolejového vozidla bude tedy nejvíce obsažena v hodnotách a_x, zatímco hodnota a_y bude úměrná převážně dostředivému zrychlení v zatáčkách. Vzhledem k velmi malé dynamice klesání a stoupání, bude složka a_z prakticky odpovídat tíhovému zrychlení. V oblasti vyšších kmitočtů pak bude právě složka a_z asi nejvíce ovlivňována vibracemi generovanými během jízdy.

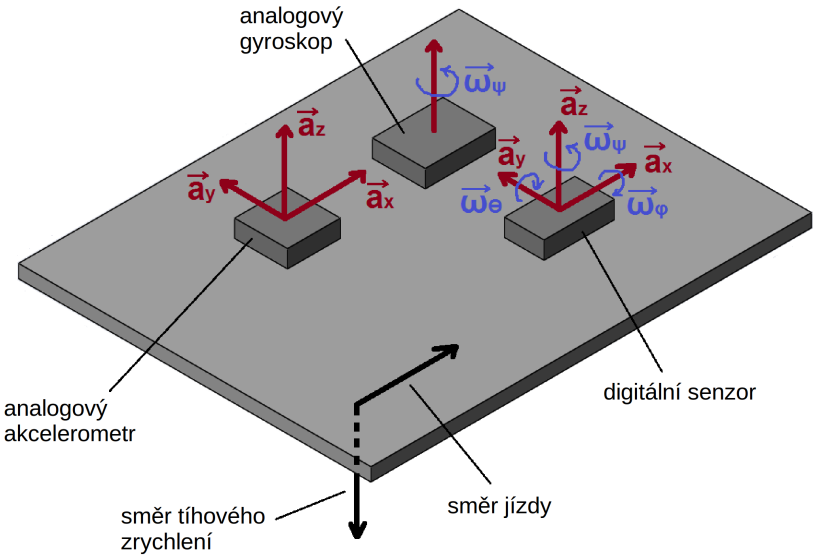
Druhým užitým typem senzoru je digitální MEMS gyroskop, jehož výstupem jsou opět tři číselné hodnoty, které odpovídají úhlovým rychlostem, kterými je s čipem senzoru otáčeno. Z hlediska terminologie by se jednalo o pohyby klonění, klopení a bočení (v angličtině roll, pitch a yaw). Při vhodné orientaci obvodu vůči vozidlu pak právě bočení obsahuje informaci o otáčení vozidla kolem svislé osy, což je z hlediska vlaku nejvýznamnější složka a zbylé dvě se u kolejových vozidel zpravidla neuplatňují v takové míře. Tato informace také umožňuje identifikovat zda vlak projíždí zatáčkou či se pohybuje po rovném úseku.

Mimo výše popsané tříosé digitální senzory, které jsou využívány pro výpočty inerciální navigace, je navržený systém vybaven navíc ještě jedním tříosým akcelerometrem a jedním jednoosým gyroskopem. Tyto senzory mají analogový výstup, takže z nich lze získávat surové, nefiltrované hodnoty s výrazně vyšším vzorkovacím kmitočtem, než je tomu u digitálních senzorů. Analogový akcelerometr je orientován shodně s digitálním a gyroskop je orientován tak, aby pokud možno nejvíce měřil úhlovou rychlost bočení (yaw). Tyto senzory jsou navigačním algoritmem využívány především k detekování zastavení, jízdy

[9]Linearita senzoru, respektive chyba linearity je jedním z parametru tohoto typu senzorů.

po rovném úseku a zahájení, trvání a ukončení průjezdu obloukem.

Pro představu o rozmístění a orientaci senzorů použitých v popisované aplikaci, je na obrázku 2.4 uveden schematický nákres. Senzory jsou umístěny tak, aby směr zrychlení a_x odpovídal co nejvíce směru jízdy vozidla a hodnota a_z směru tíhového zrychlení. Za tohoto předpokladu pak úhlová rychlost ω_φ popisuje klonění (roll), ω_θ klopení (pitch) a ω_ψ bočení (yaw).



Obr. 2.4: Uspořádání a orientace senzorů v popisované aplikaci

2.2.2 Úprava naměřených dat

Úpravou naměřených dat v rané fázi jejich zpracování je myšleno především potlačení různých rušení a chyb. Některé chyby není možné na úrovni vstupních signálů odstranit a jejich eliminace je možná pouze vhodnými algoritmy ve vyšší úrovni navigačního systému. Jiné chyby, jako například mechanické rušení a různé vibrace je možné efektivně odstranit vhodnou filtrací vstupních dat. Také ovšem existuje celá řada chyb, které je možné odstranit jen velmi obtížně nebo vůbec. S těmito je pak nutné počítat a zvážit jejich vliv na celkovou funkci zařízení.

2.2.2.1 Chyby a rušení v měřených datech

Chyby jako offset, nelinearita nebo teplotní závislost souvisí především s technologií MEMS a konkrétním typem senzorů. Konkrétní zvolené senzory a jejich parametry budou uvedeny později, během popisu hardwarové části projektu. Odstranění nebo potlačení jejich chyb je někdy zcela nemožné, jindy je lze kompenzovat na úrovni navigačního algoritmu. Další chyby vznikají nepřesností mechanické konstrukce, kdy jednotlivé osy ve kterých senzory měří nejsou orientovány přesně v předpokládaných směrech. Gyroskopické senzory také ovlivňuje fakt, že nejsou umístěny ve středu otáčení. Tyto chyby není

možné odstranit na úrovni signálů, ale lze je většinou efektivně odstranit použitím vhodných algoritmů, jak bude ukázáno později. Posledním typem chyby je mechanické rušení, které se superponuje na měřený signál a může jej značně znehodnocovat. Zdroje těchto rušení mohou být různorodé v závislosti na konkrétní aplikaci. U kolejových vozidel jsou asi nejvýznamnější zdroje mechanického rušení následující:

- Nerovnosti na povrchu kolejnic a nepřesnosti v jejich napojení
- Rázy během rozjezdu a brzdění
- Kmitání vozidla způsobené odpružením

Nerovnosti na povrchu kolejnic a nepřesnosti v jejich napojení způsobují vibrace jejichž kmitočet je přímo úměrný rychlosti jízdy a amplituda je dána rozměrem nerovnosti a dynamikou pohybu (kromě rychlosti zde hraje roli také hmotnost vozidla). Toto rušení se trvale promítá zejména do složky tíhového zrychlení a_z a částečně také do a_y. V místech nepřesného napojení kolejnic se pak přičítají různé rázy hlavně do složky a_y. Rázy během rozjezdu a brzdění se promítají především do složky zrychlení a_x a jsou velmi proměnlivé, v závislosti na konkrétním vozidle a způsobu jízdy. Kmitání vozidla způsobené odpružením ovlivňuje všechny složky zrychlení a kývavé pohyby se promítají i do výstupu gyroskopických senzorů. Míra tohoto rušení závisí na stavu a typu odpružení vozidla a množství energie, které soustava odpružení musí utlumit.

2.2.2.2 Návrh číslicového filtru

Z výše popsaného plyne, že výstupní hodnoty senzorů je nezbytné dobře filtrovat, a tak co nejvíce potlačit rušivé složky. Zvolené digitální senzory již za tímto účelem obsahují integrované filtry, které lze částečně konfigurovat. V případě analogového akcelerometru a gyroskopu byla zvolena možnost číslicové filtrace, hlavně z důvodu snadnější implementace a možnosti filtr kdykoli překonfigurovat nebo nahradit jiným.

Předpokladem úspěšné číslicové filtrace, která splní vzorkovací teorém a nebude při ní tedy docházet k aliasingu, je omezení šířky pásma ještě před vzorkováním ADC převodníkem. Aby nebyly nároky na analogový, anti-aliasingový filtr příliš vysoké a vyhověl zde i obyčejný RC článek, je využívána metoda převzorkování. Zlomový kmitočet analogového filtru 1. řádu byl zvolen přibližně 14,5 Hz, přičemž vzorkovací kmitočet je 10 kHz. Takto převzorkované hodnoty jsou filtrovány navrženým FIR filtrem a následně decimovány faktorem 128 na výsledný kmitočet 78,125 Hz. Nová data jsou tedy k dispozici každých 12,8 ms, ale protože celý cyklus měření a zápisu dat probíhá s periodou 100 ms jsou hodnoty ještě průměrovány metodou MAV přes tři vzorky.

Jako vhodné řešení filtrace a decimace se nabízelo použití polyfázového číslicového filtru [14], který v sobě spojuje jak FIR filtraci, tak i decimaci. Rovnice (2.1) představuje číslicový filtr typu FIR, kde $\boldsymbol{x}[\,]$ je vstup, $\boldsymbol{y}[\,]$ je výstup, $\boldsymbol{b}[\,]$ jsou koeficienty filtru a K je

řád filtru. Rovnice (2.2) pak vyjadřuje decimaci s faktorem M. Polyfázový filtr kombinuje oba tyto procesy, čemuž odpovídá výsledná rovnice (2.3).

$$\boldsymbol{y}\left[n\right] = \sum_{k=0}^{K} \left(\boldsymbol{b}\left[k\right] \cdot \boldsymbol{x}\left[n-k\right]\right) \tag{2.1}$$

$$\boldsymbol{y}\left[nM\right] = \sum_{k=0}^{K} \left(\boldsymbol{b}\left[k\right] \cdot \boldsymbol{x}\left[nM-k\right]\right) \tag{2.2}$$

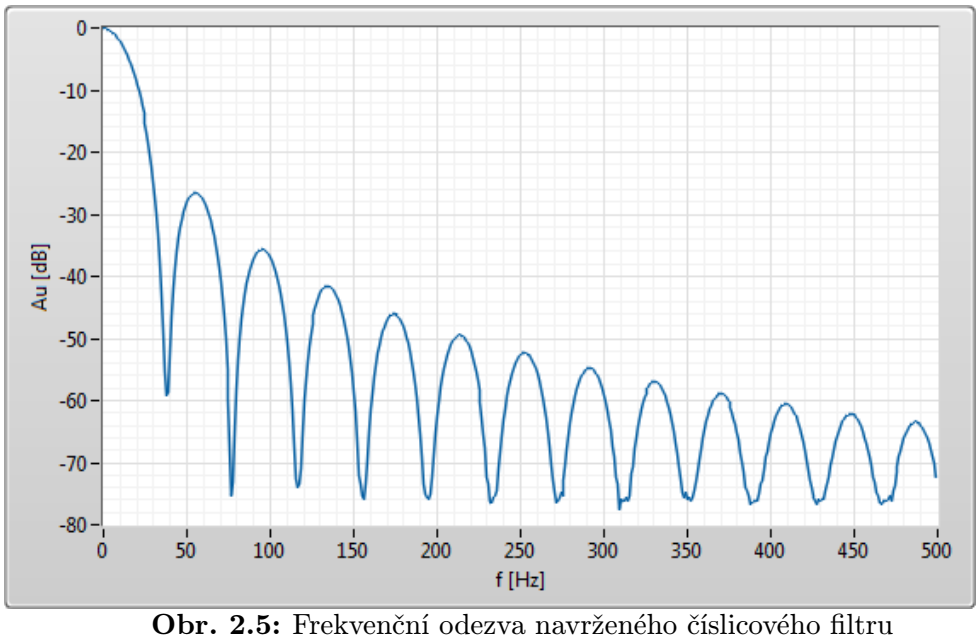
$$\boldsymbol{y}\left[n\right] = \sum_{m=0}^{M-1} \sum_{k=0}^{K} \left(\boldsymbol{b}\left[kM+m\right] \cdot \boldsymbol{x}\left[(n-k)\cdot M - m\right]\right) \tag{2.3}$$

Koeficienty FIR filtru byly spočítány v programu Matlab pro filtr typu dolní propust se zlomovým kmitočtem 16 Hz a aproximací Butterworth. Implementace v jazyku C byla založena na koeficientech celočíselného typu a zvolená implementace je blíže popsána v [15]. Výsledné parametry navrženého filtru jsou uvedené v tabulce 2.1.

Tab. 2.1: Parametry navrženého číslicového filtru

Parametr	**Hodnota**
Vzorkovací kmitočet f_s	10 kHz
Zlomový kmitočet f_c	16 Hz
Řád FIR filtru K	8
Faktor decimace M	128
Bitový rozsah	16 bitů

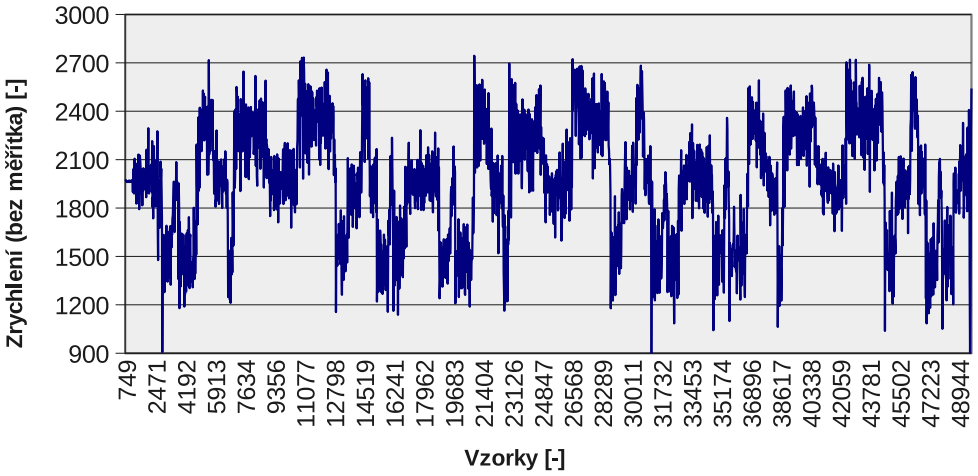
Pro další ověření funkce filtračního algoritmu, byla jeho implementace v C přeložena pro PC jako funkce v DLL knihovně. V prostředí LabVIEW pak byla tato funkce volána nad generovaným vzorkem dat a tímto způsobem byla změřena skutečná frekvenční odezva filtru. Výsledek je uveden v grafu na obrázku 2.5.



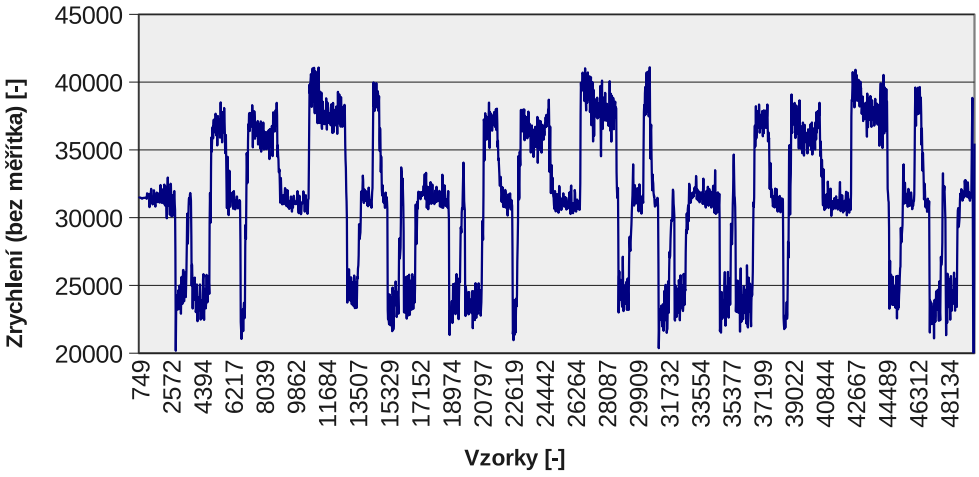
Obr. 2.5: Frekvenční odezva navrženého číslicového filtru

Navržený filtr byl také testován v časové oblasti a porovnáván s analogovým filtrem topologie Sallen-Key. Ten byl navržen jak pro porovnání číslicového filtru, tak i pro vyzkoušení jejich vzájemné kombinace, která ale nakonec nebyla implementována. Jelikož jde o dobře známou konstrukci filtru, bude jeho popis stručný. Filtr je typu dolní propust a byl spočítán pro zlomový kmitočet 16 Hz. Výsledky porovnání obou filtrů jsou uvedeny na obrázku 2.6.

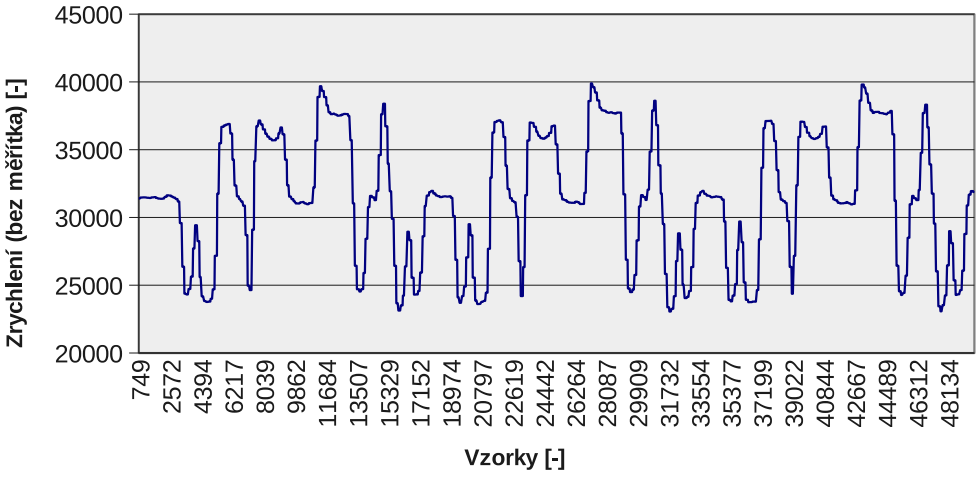
Porovnáme-li surová data z grafu na obrázku 2.6 a) s daty navzorkovanými po filtraci analogovým filtrem na obrázku 2.6 b), je vidět značné zlepšení, ale výsledek nedosahuje kvality číslicové filtrace, která je uvedena na obrázku 2.6 c). Výhodou analogového filtru je ale bezesporu absence zpoždění měřených dat, které lze při pozorném porovnání grafů u číslicového filtru pozorovat. Z výsledků je vidět, že číslicová filtrace splnila dostatečně požadavky a výsledná data jsou použitelná i přes určité zpoždění, který FIR filtr ze svého principu produkuje.



(a) Data bez filtrace



(b) Data filtrovaná analogovým filtrem Sallen-Key



(c) Data filtrovaná navrženým číslicovým filtrem

Obr. 2.6: Příklady naměřených a filtrovaných dat

2.2.3 Odvozené veličiny a výpočty

Filtrovaná data z akcelerometrů a gyroskopů jsou připravena pro aplikaci určitých výpočtů, na jejichž základě lze z původních měřených veličin a jejich kombinace odvodit další veličiny, důležité pro navigaci. Jedním ze základních výpočtů je integrace v čase, jejímž použitím lze například ze zrychlení získat rychlost a následně ujetou vzdálenost, případně z úhlové rychlosti úhel o který se vozidlo otočilo. Druhou skupinu výpočtů tvoří různé transformace souřadných systému v trojrozměrném prostoru, s jejichž pomocí lze například implementovat inerciální systém typu strap-down.

2.2.3.1 Integrace zrychlení a úhlové rychlosti

Hodnoty získané z akcelerometrů a senzorů typu rate-gyro odpovídají, po určité úpravě rozsahu a měřítka, fyzikálním veličinám zrychlení $a\,[m \cdot s^{-2}]$ a úhlové rychlosti $\omega[rad \cdot s^{-1}]$. Odtud je postup k dalším veličinám jako je rychlost a poloha teoreticky jednoduchý, neboť první integrací zrychlení v čase získáme rychlost a druhou integrací pak ujetou vzdálenost. Uvažujeme-li konstantní zrychlení, platí pro výpočet rychlosti a následně i vzdálenosti rovnice (2.4), kde d je vzdálenost, v rychlost, a zrychlení a b je chybová složka zrychlení (bias senzoru). Jelikož se pohybujeme v číslicovém prostředí, má větší význam diskrétní varianta výpočtu uvedená v rovnici (2.5). [16] až [18]

$$\vec{d}\,(t) \;= \int_0^t \vec{v}\,(t)\,dt = \int_0^t ((\vec{a}+\vec{b})t + \vec{v_0})dt = \frac{1}{2}(\vec{a}+\vec{b})t^2 + v_0 t + \vec{d_0} \tag{2.4}$$

$$\vec{d_k} = \frac{1}{2}(\vec{a}_{k-1} + \vec{b}_{k-1})\,(t_k - t_{k-1})^2 + \vec{v}_{k-1}\,(t_k - t_{k-1}) + \vec{d}_{k-1} \tag{2.5}$$

Právě chybová složka zde představuje hlavní problém praktické implementace uvedených vztahů, protože se neustále integruje společně se zrychlením a i po relativně krátkém časovém intervalu je vypočítaná hodnota velmi vzdálená realitě. Tato rušivá složka obsahuje především chyby vlastního senzoru, jako je offset, nelinearita nebo šum[10]. Další chyba je způsobena nepřesnou montáží senzoru vůči vozidlu, kdy se dopředná složka zrychlení, popřípadě i příčná (dostředivá) rozkládají i do ostatních os. Totéž platí pro tíhové zrychlení, které se vlivem naklánění vozidla rozkládá částečně i do os x a y. Zatímco chyby spojené s montáží senzoru a působením tíhového zrychlení lze poměrně efektivně odstranit vhodným algoritmem, který bude popsán později, vlastní chyby senzoru lze odstranit jen obtížně. Provedením měření v klidovém stavu, uložením hodnot a následnou kompenzací, lze chyby na určitou dobu potlačit, nicméně i malá chyba se neustále integruje a roste.

V případě rate-gyroskopu lze za odvozenou veličinu považovat úhel, který je možné získat integrací úhlové rychlosti v čase. Je-li úhlová rychlost konstantní pak platí rovnice

[10] Detailnější rozbor chyb senzorů je uveden například v [19].

(2.6) a (2.7), kde φ je úhel, ω je úhlová rychlost a b je chybová složka úhlové rychlosti. [16] až [18]

$$\varphi(t) = \int_0^t (\omega + b)\, dt = (\omega + b)\, t + \varphi_0 \tag{2.6}$$

$$\varphi_k = (\omega_{k-1} + b_{k-1})\,(t_k - t_{k-1}) + \varphi_{k-1} \tag{2.7}$$

2.2.3.2 Aplikace strap-down systému a transformace

Výpočtem, typickým zejména pro strap-down systémy, je transformace souřadného systému senzoru do navigačního souřadného systému. V popisovaném případě se jedná o navigační souřadný systém orientovaný vůči tíhovému zrychlení. Běžné uspořádání strap-down systému obsahuje 3D akcelerometr a 3D gyroskop a data z akcelerometru jsou transformována na základě úhlových rychlostí z gyroskopu. Základní rovnice pro transformaci souřadnic mezi dvěma kartézskými systémy je vyjádřen rovnicí (2.8) [13], kde $\boldsymbol{v}$ je obecný vektor, $\boldsymbol{v}'$ je tentýž vektor v novém souřadném systému a $\boldsymbol{C}$ je transformační matice

$$\boldsymbol{v}' = \boldsymbol{C} \times \boldsymbol{v} \tag{2.8}$$

Pro kontinuální přepočet vektoru zrychlení $\boldsymbol{a}$ na $\boldsymbol{v}'$ je třeba znát časový průběh transformační matice $\boldsymbol{C}(t)$. Další výpočet vychází z rovnice (2.9) [19], která vyjadřuje změnu transformační matice v čase (derivaci $\boldsymbol{C}$).

$$\frac{d\boldsymbol{C}}{dt} = \lim_{\triangle t \to 0} \frac{\boldsymbol{C}(t + \triangle t) - \boldsymbol{C}(t)}{\triangle t} \tag{2.9}$$

Matici $\boldsymbol{C}(t + \triangle t)$ lze dále vyjádřit jako součin původní matice $\boldsymbol{C}(t)$ a matice $\boldsymbol{A}(t)$, která vyjadřuje rotaci systému mezi časem t a $(t+\triangle t)$. Za použití aproximace pro malé úhly [19] pak platí rovnice

$$\boldsymbol{A}(t) = \boldsymbol{I} + \triangle\boldsymbol{\Psi} \tag{2.10}$$

kde $\boldsymbol{I}$ je jednotková matice a $\triangle\boldsymbol{\Psi}$ je

$$\triangle\boldsymbol{\Psi} = \begin{bmatrix} 0 & -\triangle\psi & \triangle\theta \\ \triangle\psi & 0 & -\triangle\varphi \\ -\triangle\theta & \triangle\varphi & 0 \end{bmatrix} \tag{2.11}$$

Po dosazení do rovnice (2.9) a provedení substituce dostaneme

$$\begin{aligned}\frac{d\boldsymbol{C}}{dt} &= \lim_{\triangle t\to 0}\frac{\boldsymbol{C}(t+\triangle t)-\boldsymbol{C}(t)}{\triangle t}\\ &= \lim_{\triangle t\to 0}\frac{\boldsymbol{C}(t)\boldsymbol{A}(t)-\boldsymbol{C}(t)}{\triangle t}\\ &= \lim_{\triangle t\to 0}\frac{\boldsymbol{C}(t)(\boldsymbol{I}+\triangle\boldsymbol{\Psi})-\boldsymbol{C}(t)}{\triangle t}\\ &= \boldsymbol{C}(t)\lim_{\triangle t\to 0}\frac{\triangle\boldsymbol{\Psi}}{\triangle t}\end{aligned} \tag{2.12}$$

Změna úhlu za čas je v limitě rovna derivaci úhlu v čase, což představuje definici úhlové rychlosti a lze tedy napsat

$$\frac{d\boldsymbol{C}}{dt} = \boldsymbol{C}(t)\boldsymbol{\Omega}(t) \tag{2.13}$$

kde

$$\boldsymbol{\Omega}(t) = \begin{bmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix} \tag{2.14}$$

Řešením diferenciální rovnice (2.13) je exponenciální funkce a její implementace je možná pomocí Taylorova rozvoje. [19] Rovnice (2.15) [19] je pak výsledným řešením časového průběhu[11] transformační matice.

$$\boldsymbol{C}(t+\triangle t) = \boldsymbol{C}(t)\left(\boldsymbol{I}+\frac{\sin\sigma}{\sigma}\boldsymbol{B}+\frac{1-\cos\sigma}{\sigma^2}\boldsymbol{B}^2\right) \tag{2.15}$$

kde σ je velikost vektoru úhlových rychlostí $\boldsymbol{\omega}$ násobeného časem, tedy

$$\sigma = |\boldsymbol{\omega}\cdot\triangle t| \tag{2.16}$$

a matice $\boldsymbol{B}$

$$\boldsymbol{B} = \begin{bmatrix} 0 & -\omega_z\triangle t & \omega_y\triangle t \\ \omega_z\triangle t & 0 & -\omega_x\triangle t \\ -\omega_y\triangle t & \omega_x\triangle t & 0 \end{bmatrix} \tag{2.17}$$

Tento výsledek je již možné implementovat do inerciálního algoritmu, protože úhlové rychlosti vektoru $\boldsymbol{\omega}$ jsou výstupem rate-gyroskopu a $\triangle t$ je vzorkovací perioda měření. Mimo popsanou metodu založenou na směrových kosinech (direction cosines) existují ještě další postupy jako metoda Eulerových úhlů nebo kvaterniony. [19]

Strap-down algoritmus založený na datech z gyroskopických senzorů tak, jak byl právě popsán, není jedinou součástí popisovaného navigačního systému pro kolejová vozidla a transformace souřadnic je zde prováděna ještě dalšími algoritmy, které budou popsány později. Proto, s ohledem na nižší náročnost implementace, byla zvolena metoda směrových kosinů a celý algoritmus je využíván jako zpřesňující nástroj.

[11] Časový průběh v diskrétní formě času, kdy jednotlivé vzorky přichází s periodou $\triangle t$.

2.3 Navigační úloha

Nejčastější řešení navigační úlohy v dnešních komerčních zařízeních spočívá v integraci dvou výše popsaných komplexnějších celků, a sice IMU a GNSS přijímače, jejichž výstupy jsou zpracovávány algoritmem Kalmanovy filtrace. Výsledkem je poměrně přesná informace o poloze navigovaného objektu a jeho orientaci v prostoru, jak ve chvíli příjmu GNSS signálu, tak i po určitý čas bez příjmu. Estimaci polohy během výpadku nebo zhoršení příjmu GNSS zajišťuje právě Kalmanova filtrace, jejíž stavový vektor obsahuje všechna navigační data z IMU, GNSS nebo případně z dalších senzorů. [20], [21]

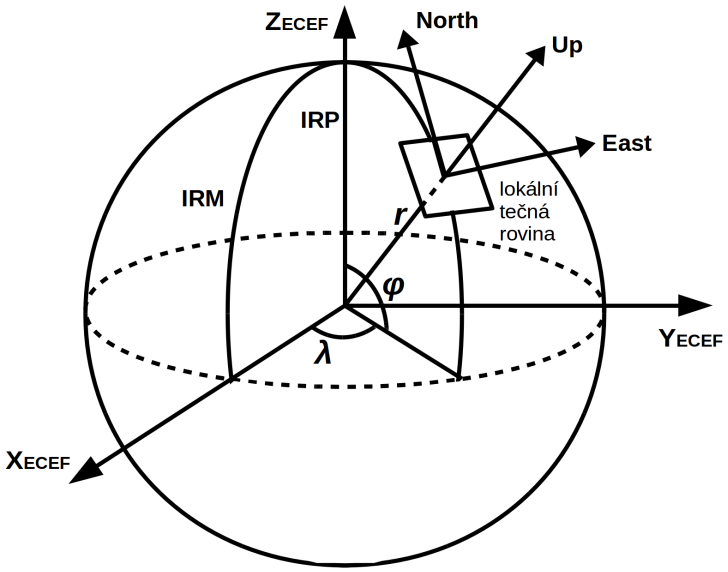
Výstupem IMU jsou zpravidla tři hodnoty zrychlení a tři hodnoty úhlové rychlosti orientované vůči souřadnému systému navigovaného objektu, případně data z dalších senzorů, jako je magnetometr, rychloměr, výškoměr apod. Navigační data z GNSS přijímače jsou obvykle ve formě zeměpisné šířky a délky a nadmořské výšky a jsou vztažena k souřadnému systému WGS84 [9]. Aby bylo možné provádět navigační výpočty, například za pomoci Kalmanovy filtrace, je nutné GNSS data matematicky přepočítat (transformovat) ze systému WGS84 do lokální referenční soustavy s definovaným počátkem. Pro použití ve stavovém vektoru navigačního algoritmu byla standardizována reprezentace ve formátu North East Down (NED), respektive East North Up (ENU). [21]

Jako mezikrok transformace WGS84 je používána reprezentace polohy v souřadném systému ECEF (Earth-Centered, Earth-Fixed). Tento souřadný systém je vlastně kartézská reprezentace dat ve formátu WGS84, protože stejně jako WGS84, má ECEF počátek v těžišti Země. Osa z směřuje k mezinárodnímu referenčnímu pólu (IRP)[12], osa x směřuje do průsečíku nulté rovnoběžky (rovníku) a mezinárodního referenčního poledníku (IRM)[13] a osa y je kolmá na obě předchozí osy a tedy podle pravidla pro pravotočivé kartézské souřadné systémy směřuje na východní polokouli. [22]

Vztah mezi systémy WGS84, ECEF a NED, resp. ENU je dobře patrný z obrázku 2.7. Úhly φ a λ představují zeměpisnou šířku a délku a společně s ramenem r, které lze vypočítat na základě znalosti výšky vztažené k referenčnímu elipsoidu WGS84 a matematické rovnice tohoto elipsoidu, tvoří reprezentaci v systému WGS84. Tyto polární rozměry pak lze přepočítat do systému ECEF, který je znázorněn osami X_{ECEF}, Y_{ECEF} a Z_{ECEF}. Souřadný systém NED/ENU je projekcí ECEF do lokální tečné roviny v daném bodě a je znázorněn osami North, East a Up, resp. Down. [22]

[12] Osa není totožná s osou otáčení Země. [22]

[13] IRM leží 5,31 úhlových vteřin východně od nultého poledníku. [9]



Obr. 2.7: Vztah mezi souřadnými systémy WGS84, ECEF a NED/ENU [22]

Transformace mezi ECEF a NED je vyjádřena rovnicí (2.18) [22], kde $\boldsymbol{P}_{NED}$ je vektor polohy v souřadném systému NED, $\boldsymbol{P}_{ECEF}$ je vektor polohy v souřadném systému ECEF, $\boldsymbol{P}_{REF}$ je vektor polohy počátku souřadného systému NED v souřadném systému ECEF (poloha tečné roviny) a $\boldsymbol{R}$ je transformační matice. Transformační matice je pak vyjádřena vztahem (2.19) [22], kde φ a λ představují zeměpisnou šířku a délku.

$$\boldsymbol{P}_{NED} = \boldsymbol{R}^T(\boldsymbol{P}_{ECEF} - \boldsymbol{P}_{REF}) \tag{2.18}$$

$$\boldsymbol{R} = \begin{bmatrix} -\sin\varphi\cos\lambda & -\sin\lambda & -\cos\varphi\cos\lambda \\ -\sin\varphi\sin\lambda & \cos\lambda & -\cos\varphi\sin\lambda \\ \cos\varphi & 0 & -\sin\varphi \end{bmatrix} \tag{2.19}$$

Jak již bylo zmíněno v úvodu této kapitoly, popsané řešení navigační úlohy za pomoci IMU, GNSS a estimačního algoritmu je dnes asi nejrozšířenější způsob, jak realizovat různá navigační zařízení, ať už se jedná o osobní, dopravní či průmyslové aplikace. Hledání výkonnějšího algoritmu implementujícího Kalmanovu filtraci tak, jak bylo popsáno výše, není přímo předmětem této práce, která se zaměřuje na navigační úlohu z trochu odlišného úhlu pohledu, ale znalost výše popsané navigační úlohy, souřadných systémů a jejich transformací, GNSS systémů a IMU je nezbytná pro jakoukoli výzkumnou činnost a vývoj v tomto oboru.

2.4 Mapy a projekce

Navigační aplikace určené pro osobní použití, např. navigace v automobilu, se neobejdou bez grafické reprezentace získaných údajů. Aby je mohl člověk snadno přečíst a využít, je

poloha navigovaného objektu, včetně jeho orientace, promítnuta na určitý mapový podklad, například na mapu městské části, kterou vozidlo právě projíždí. V dnešní době již existuje celá řada softwarových mapových podkladů, z nichž je většina volně přístupná veřejnosti a zahrnuje dopravní, urbanistické i fotografické mapy prakticky celého světa. Mapy jsou dostupné na webových serverech a mezi nejznámější u nás patří mapy od společnosti Google [23], komunitní mapy OpenStreetMap (OSM) [24], Mapy.cz provozované společností Seznam [25] a případně pak přesné geodetické mapy dostupné na webu ČÚZK [26].

2.4.1 Zobrazení

Aby bylo možné prostorová data z GNSS systému, tedy ve formátu WGS84, situovat do dvourozměrné mapy, je nutné použít některé matematické zobrazení. Jedním z možných zobrazení je Mercatorovo, resp. modernizovaná elektronická forma Web Mercator. Toto zobrazení vychází z klasické Mercatorovy projekce, která promítá kulovou nebo eliptickou plochu na plochu válcovou s osou totožnou se zemskou osou. Projekce tedy udržuje vzájemnou kolmost poledníků a rovnoběžek (konformní zobrazení), ale značně zkresluje rozměry směrem k pólům. Projekce Web Mercator je uzpůsobena pro webové a počítačové aplikace a implementuje navíc možnost přiblížení (zoom). [27]

Standardizaci zobrazení používaných v on-line mapách zajišťuje Mezinárodní asociace producentů ropy a zemního plynu (IOGP), resp. její součást European Petroleum Survey Group (EPSG) [28]. První standard EPSG:3785 „Popular Visualization Mercator“ využívá projekci z kulové plochy, zatímco data z GNSS jsou referencována vůči elipsoidu WGS84. To způsobuje zkreslení, které může dosáhnout, v závislosti na zeměpisné šířce, i několik desítek metrů. Druhý standard, EPSG:3857 „WGS84/Pseudo-Mercator“ již využívá projekci z elipsoidu WGS84 a je tedy v zobrazení GNSS dat přesnější. Tento posledně jmenovaný standard používají prakticky všechny populární mapové servery jako Google maps, OSM nebo Mapy.cz. [23] až [25]

2.4.2 Mapové podklady

V popisované aplikaci navigačního systému pro kolejová vozidla hraje mapa a mapový podklad jen informativní roli a pro zjednodušení jsou využívány mapové podklady OSM, dle výše uvedeného standardu EPSG:3857. Bohužel, vzhledem k první verzi standardu s kulovou plochou namísto elipsoidu a celkových nejasností kolem odpovědnosti za standard[14], není ve většině zemí povoleno použití výše uvedených mapových podkladů pro legislativní, geodetické a jiné oficiální účely. [29] Tato skutečnost by mohla ovlivnit i popisovaný systém, v závislosti na konkrétní aplikaci a související legislativě. Pak by bylo vhodné používat schválené mapové podklady, například pro evropské státy uvedené v [30].

[14] Jako informační zdroje pro standard EPSG:3857 uvádí EPSG společnosti Google a Microsoft. [28]

V případě ČR jsou přesné geodetické mapy tvořeny v projekci S-JTSK/Krovak [31] a jsou dostupné na webu ČÚZK. [26]

Hlavní činnost popisovaného navigačního systému z hlediska práce s mapou je detekování vstupu a výstupu navigovaného železničního vozidla do nebo z určitého definovaného prostoru. Za tímto účelem je nutné umět přepočítat jeden stupeň zeměpisné šířky a délky na vzdálenost v metrech tak, aby následně mohlo být provedeno měření v mapě, resp. porovnání s tabulkou bodů definovaných v mapě. Přepočet úhlu na vzdálenost vychází z navigačních tabulek [32] a platí rovnice (2.20) [33] a (2.21) [33],

$$d_m = m_1 + m_2 \cdot \cos(2\varphi) + m_3 \cdot \cos(4\varphi) + m_4 \cdot \cos(6\varphi) \tag{2.20}$$

$$d_p = p_1 \cdot \cos(\varphi) + p_2 \cdot \cos(3\varphi) + p_3 \cdot \cos(5\varphi) \tag{2.21}$$

kde d_m je délka jednoho stupně zeměpisné šířky v metrech, d_p je délka jednoho stupně zeměpisné délky v metrech, φ je zeměpisná šířka ve stupních a koeficienty m_1 až m_4 a p_1 až p_3 jsou konstanty vypočítané pro elipsoid WGS84. Pro úplnost jsou tyto konstanty uvedeny v tabulce 2.2.

Tab. 2.2: Konstanty pro přepočet jednoho stupně zeměpisné šířky a délky na vzdálenost [33]

Konstanta	**Hodnota**
m_1	111132,92
m_2	-559,82
m_3	1,175
m_4	-0,0023
p_1	111412,84
p_2	-93,5
p_3	0,118

3

Určování polohy kolejových vozidel

Navigační úloha kolejových vozidel je aktuální a významná problematika, která je řešena v odborných skupinách prakticky po celém světě. Celá tato problematika obsahuje mimo technickou část také velmi rozsáhlou legislativní a případně i realizační oblast a je tedy jako celek značně nad rámec tohoto textu. Zde je řešena pouze část technická, kde lze hledat nové, inovativní postupy, neobvyklé kombinace senzorů, odlišné zpracování dat a podobně, což může někdy přinést zajímavé výsledky.

Navigační systém, který je předmětem této práce, využívá IMU typu strap-down, s integrovanými senzory technologie MEMS a také GNSS přijímač. Navigační úloha zde není řešena obecným způsobem dlouhodobého určování přesné polohy, jak bylo popsáno v předchozí kapitole, ale spíše je orientována na verifikaci projížděné trati, určování rychlosti, režimu provozu a jiných dispečerských dat. Kolejová vozidla také představují specifickou oblast pro navigační úlohu proto, že jejich poloha je pevně spojena s traťovým svrškem a v podstatě jde o určování polohy vozidla na předem dané trati. To ale neznamená, že popsané principy inerciální navigace, zpracování dat a projekce do mapy nejsou využívány.

Pohyb kolejových vozidel má také určitá specifika[1], která lze využít v rámci navigačního algoritmu ke zpřesnění či zjednodušení výpočtů. Mezi tyto specifické vlastnosti lze zahrnout následující body:

- Pohyb kolejových vozidel je zpravidla plynulý s relativně nízkou dynamikou
- Úhly klonění φ a klopení θ se mění velmi málo a pozvolně
- Otáčení (změna úhlu ψ) je plynulé a poloměr oblouku je velký
- Pohyb je omezen kolejovým svrškem na definovaný počet možných trajektorií
- Vozidlo zastavuje ve stanicích, na zastávkách nebo v jiných definovaných prostorech

Na základě uvedených bodů je zřejmé, že například podélná a příčná složka zrychlení jsou pro výpočty důležitější a svislou složku lze použít pro přepočet souřadného systému a dále

[1] Myšleno především v železniční dopravě, linkách tramvají a metra. V případě kolejových vozidel provozovaných v dolech, zábavních parcích apod., tato specifika platit nemusí.

zanedbat. Z hlediska detekce charakteristických segmentů, která bude popsána později je důležitá úhlová rychlost ω_ψ a její přesnější měření je tedy důležité. Pravidelné zastávky jsou využívány ke kompenzaci a potlačení integrovaných chyb atd.

3.1 Transformace souřadného systému

Přepočet vektoru zrychlení ze souřadného systému senzorů, resp. měřicího zařízení, do navigační souřadné soustavy byl již v podstatě popsán výše, v rámci části věnované strap-down algoritmu. Stejně tak byla definována i navigační soustava, ve které popisovaný systém pracuje. Mimo tyto obecné principy používané v navigačních algoritmech, lze u kolejových vozidel využít některé další specifické situace k zpřesnění navigačních výpočtů.

3.1.1 Přepočet do soustavy orientované ve směru tíhového zrychlení

Pokud je vozidlo v klidu, rozkládá se tíhové zrychlení do složky měřeného zrychlení a_z, ale také částečně do složek a_x a a_y. To je způsobeno několika faktory, zejména nepřesnou montáží senzoru v měřicím zařízení, nepřesnou orientací zařízení vůči souřadnému systému vozidla a také tím, jak je vozidlo nakloněné vůči rovině kolmé na tíhové zrychlení (vodováze). Náklon vozidla je charakterizován úhly φ (klonění) a θ (klopení), které lze spočítat z vektoru zrychlení podle rovnic (3.1) [13] a (3.2) [13], kde φ je úhel klonění, θ úhel klopení a a_x, a_y a a_z jsou složky vektoru zrychlení.

$$\varphi = \arctan\left(\frac{a_y}{a_z}\right) \tag{3.1}$$

$$\theta = \arctan\left(\frac{-a_x}{\sqrt{(a_y^2 + a_z^2)}}\right) \tag{3.2}$$

Obecnou rotaci v trojrozměrném prostoru lze složit ze tří elementárních rotací kolem každé osy provedených ve správném pořadí. [13] V případě úhlu φ je to rotace kolem osy x a v případě úhlu θ rotace kolem osy y. Výsledná matice rotace $\boldsymbol{R}_{xy}$ vznikne tedy násobením elementárních rotačních matic [13] pro rotace kolem osy x a y v této podobě

$$\boldsymbol{R}_{xy} = \begin{bmatrix} \cos\theta & \sin\varphi\sin\theta & \cos\varphi\sin\theta \\ 0 & \cos\varphi & -\sin\varphi \\ -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{bmatrix} \tag{3.3}$$

Transformace vektoru zrychlení do nové soustavy orientované vůči tíhovému zrychlení je provedena již popsaným maticovým násobením transformační matice s původním vektorem, tedy platí následující rovnice

$$\begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} = \boldsymbol{R}_{xy} \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \tag{3.4}$$

kde a_x , a_y a a_z jsou složky původního vektoru, a'_x , a'_y a a'_z jsou složky nového vektoru a $\boldsymbol{R}_{xy}$ je transformační matice.

Výpočet matice rotace a přepočet vektoru zrychlení je proveden vždy na začátku po startu navigačního systému a také vždy, když vozidlo zastaví, například ve stanici. Výpočet se provádí nad určitým vzorkem naměřených dat, který se průměruje tak, aby výsledná rotační matice byla co nejpřesnější a rušivé složky, jako je například pohyb cestujících v případě tramvaje, se vyrušily. Takto spočítaná rotační matice je algoritmem využívána až do další výpočtové smyčky během následujícího zastavení. Pro udržení dostatečné přesnosti i během jízdy mezi zastávkami, je transformační matice upravována výstupem strap-down algoritmu.

3.1.2 Přepočet do soustavy orientované ve směru jízdy vozidla

Zmíněná nepřesná montáž měřicího zařízení vůči vozidlu také způsobuje, že dopředné zrychlení vyvolané pohonnou jednotkou se mimo složky zrychlení a_x také částečně projevuje ve složce a_y[2]. Ke kompenzaci této nepřesnosti jsou využívány okamžiky, kdy se vozidlo pohybuje po rovném úseku a hodnota úhlové rychlosti ω_ψ je tak pod definovanou úrovní. Pohyb musí být zároveň dostatečně dynamický, s výraznou velikostí akcelerace, např. při rozjezdu po zastavení v zastávce. Jsou-li tyto podmínky splněny, algoritmus nejprve určí úhel ψ podle rovnice (3.5) [13], kde a'_x a a'_y jsou složky zrychlení přepočítané podle rovnic (3.1) až (3.4) a ψ je úhel o který se odchyluje soustava zařízení od soustavy vozidla.

$$\psi = \arctan\left(\frac{a'_y}{a'_x}\right) \tag{3.5}$$

Následně je použita matice pro elementární rotaci kolem osy z, do které je dosazen vypočítaný úhel, čímž vznikne výsledná rotační matice $\boldsymbol{R}_z$, vyjádřená rovnicí (3.6).

$$\boldsymbol{R}_z = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6}$$

Transformace vektoru zrychlení do nové soustavy orientované vůči směru jízdy vozidla je provedena analogicky s předchozím případem rovnicí (3.7),

$$\begin{bmatrix} a''_x \\ a''_y \\ a''_z \end{bmatrix} = \boldsymbol{R}_z \begin{bmatrix} a'_x \\ a'_y \\ a'_z \end{bmatrix} \tag{3.7}$$

[2]Obecně by se projevilo i ve složce a_z, ale předpokladem je již aplikovaná transformace popsaná rovnicemi (3.1) až (3.4).

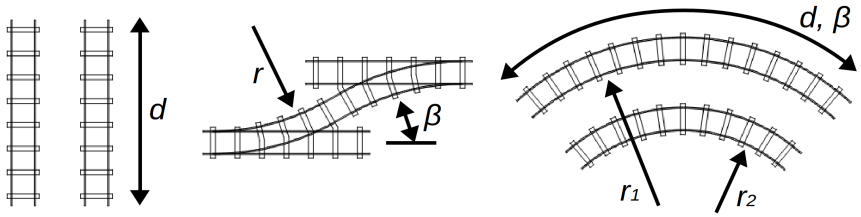
kde a'_x, a'_y a a'_z jsou složky původního vektoru, a''_x, a''_y a a''_z jsou složky nového vektoru a $\boldsymbol{R}_z$ je transformační matice.

Výpočet se opět provádí nad delším vzorkem naměřených dat, která se průměrují, aby byly potlačeny rušivé složky způsobené oscilačními pohyby vozidla vlivem odpružení a různé vibrace. Mimo okamžiky kdy je prováděn výpočet je transformační matice aktualizována výstupem strap-down algoritmu.

3.2 Detekování charakteristických segmentů tratě

Na rozdíl od výše popsaných transformací, které jsou již známou metodou pro zpřesnění inerciálních dat, je detekování charakteristických segmentů tratě jedním ze zmíněných inovativních přístupů, který byl navržen a implementován. Jelikož se jedná o jeden z hlavních přínosů celé práce, bude nyní popsán princip detekce segmentů a později, během popisu softwarové části, bude ještě přiblížena vlastní implementace.

Jak bylo naznačeno v úvodu této kapitoly, navigace, resp. určování polohy kolejových vozidel má své specifické možnosti, které u řešení obecné navigační úlohy nelze využít. Příkladem jsou určité charakteristické segmenty železniční tratě, které je možné detekovat a případně spojit s polohou vozidla na trati, neboť trasa (trajektorie) kolejového vozidla je zpravidla předem známá a plně definovaná. Příklady charakteristických segmentů tratě jsou uvedeny na obrázku 3.1. Pokud by se podařilo jednotlivé segmenty identifikovat, například na základě vypočítaného úhlu a poloměru, bylo by možné určit, jak byla přehozena právě projetá výhybka nebo po které koleji vlak projel obloukem. Tyto data by pak bylo možné spojit s údaji z GNSS, potažmo s mapou nebo schématem tratě a odeslat informaci, varování apod.



Obr. 3.1: Charakteristické segmenty železniční tratě

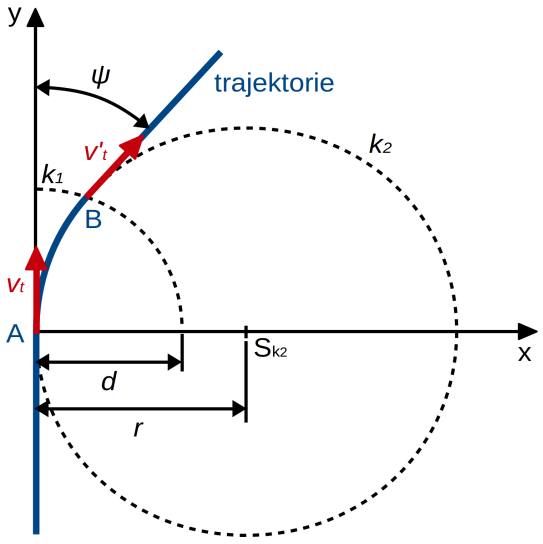
Z popisu myšlenky detekování segmentů plyne, že tento postup klade značné nároky na přesnost měřených veličin, resp. na složitost a součinnost jednotlivých výpočetních algoritmů. Vzhledem ke snaze o jednoduchost hardwaru a použití integrovaných MEMS senzorů je zřejmé, že přesnost měření IMU není vysoká a použitelný výsledek je nutné pracně dopočítávat.

3.2.1 Výpočet poloměru oblouku

Jedním z navržených a algoritmizovaných výpočtů je určování poloměru projížděného oblouku. Výpočet je založen na geometrii a předpokladu, že projížděný oblouk je skutečně částí kružnice. Pokud se jedná o utaženou nebo naopak otevřenou zatáčku (obecně s proměnným poloměrem), systém aproximuje tuto křivku částí kružnice a vrátí odpovídající poloměr. To v zásadě nebrání identifikaci segmentu a pro kolejová vozidla je tento přístup dostačující. Problém by představovala aplikace tohoto algoritmu např. pro automobilový provoz, kde je většina zatáček částí nějaké nekruhové křivky a vypočítané hodnoty by mohly být značně vzdálené realitě.

Než může algoritmus provést vlastní výpočet poloměru, je nutné nejprve detekovat počátek a následně i konec projížděného oblouku. Z tohoto důvodu je výsledek vrácen vždy až po dokončení průjezdu daným segmentem. Tato část algoritmu je tedy velmi důležitá a bude detailněji popsána v kapitole zabývající se softwarovým zpracováním naměřených dat, kde budou podstatné algoritmy detailně popsány.

Princip výpočtu je ukázán na obrázku 3.2. Vyznačená trajektorie se skládá ze dvou rovných částí, navzájem spojených obloukem, který leží na kružnici k_2 o poloměru r. Začátek oblouku je označen bodem A a konec bodem B. Celá situace je zasazena do roviny pravoúhlého souřadného systému definovaného osami x a y tak, že počátek je totožný s bodem A a osa y je tečnou k trajektorii v tomto bodě. Vektory $\vec{v}_t$ a $\vec{v}'_t$ představují tečnou složku rychlosti, kterou se vozidlo mezi body A a B pohybuje a d (poloměr kružnice k_1) je vlastně vzdálenost, o kterou se vozidlo při průjezdu obloukem posune. Přesněji řečeno, tato vzdálenost odpovídá tětivě části kružnice, která tvoří projížděný oblouk, nikoli ujeté vzdálenosti, která je rovna délce oblouku, ale pro malé úhly je tento rozdíl pod úrovní přesnosti měření a lze ho proto zanedbat. Úhel ψ je pak úhel, o který se vozidlo po průjezdu obloukem otočí vůči svislé ose.



Obr. 3.2: Princip výpočtu poloměru oblouku

Předpokladem výpočtu poloměru r je schopnost algoritmu určit body A a B, tedy počátek a konec oblouku, dále uhel otočení ψ a také vzdálenost d. To je možné díky měření úhlové rychlosti ω_ψ a složky zrychlení a_x, kdy po integraci systém získá hodnotu úhlu ψ i rychlosti v_t, z níž další integrací určí vzdálenost d. Na základě obrázku 3.2 lze kružnici k_1 vyjádřit následující rovnicí

$$x^2 + y^2 - d^2 = 0 \tag{3.8}$$

a kružnici k_2 rovnicí

$$(x - r)^2 + y^2 - r^2 = 0 \tag{3.9}$$

kde d je známá vzdálenost, r je hledaný poloměr a x a y jsou souřadnice bodu ležícího na kružnici k_1, resp. k_2. Z obrázku 3.2 je dále vidět, že bod B také představuje průsečík kružnic k_1 a k_2 a lze ho v souřadném systému definovat jako $B\,[B_x; B_y]$. Dosazením bodu B do rovnic (3.8) a (3.9) lze tyto řešit jako soustavu dvou rovnic pro dvě neznámé a vyjádřit pak souřadnice B_x a B_y následovně

$$B_x = \frac{d^2}{2r} \tag{3.10}$$

$$B_y = \sqrt{(d + \frac{d^2}{2r})(d - \frac{d^2}{2r})} \tag{3.11}$$

Omezíme-li se na 1. kvadrant roviny zobrazený na obrázku 3.2 a pouze na možná (fyzikálně reálná) řešení, musí platit, že vzdálenost d náleží intervalu $(0; \infty)$ a hledaný poloměr r je v intervalu $\left\langle \frac{\sqrt{2}}{2}d; \infty \right)$. Dále je možné vyjádřit úhel ψ v radiánech rovnicí

$$\psi = \frac{\pi}{2} - \arctan\left(-\frac{B_x - r}{B_y}\right) \tag{3.12}$$

za předpokladu, že úhel ψ náleží intervalu $\left(0; \frac{\pi}{2}\right)$. Poslední neznámou v rovnici (3.12) je tak právě hledaný poloměr r, který ale není třeba z rovnice dále za pomoci trigonometrických vzorců analyticky vyjadřovat, neboť algoritmus hledající poloměr oblouku pracuje na principu iterace. Změřený úhel ψ je porovnáván s vypočítaným úhlem, přičemž poloměr je inkrementován v rámci výše popsaného intervalu tak dlouho, než je dosažena přijatelná shoda obou úhlů.

Takto je tedy řešena matematická část algoritmu pro výpočet poloměru oblouku, ale vzhledem k omezením a předpokladům, které z výše popsaného postupu plynou, je samotná implementace výrazně složitější. Klíčové části algoritmu budou ještě detailněji popsány později, ale matematická podstata popsaná zde zůstává zachována.

3.3 Syntéza dat

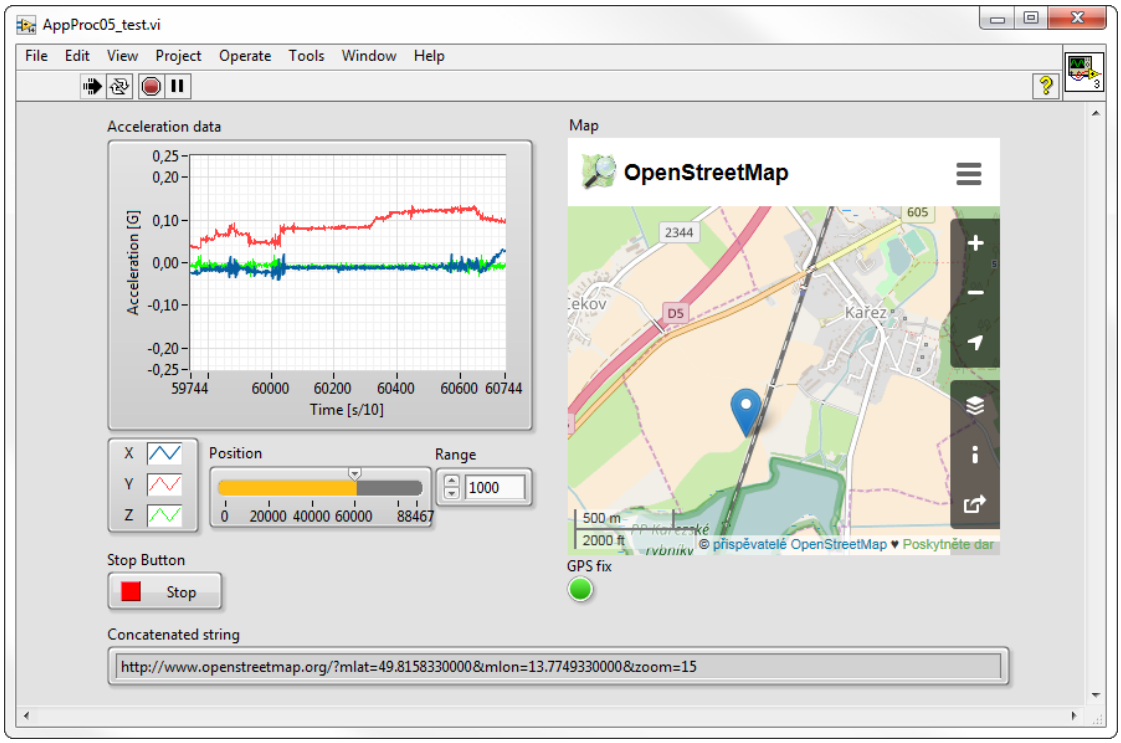
Z dosavadního popisu jednotlivých subsystémů plyne, že každý má své specifické chyby a nedostatky, které za určitých podmínek naprosto znehodnocují jeho výstup. Například GNSS přijímač trpí, zvlášť v husté výstavbě velkoměst, nedostatečným počtem viditelných satelitů, což značně snižuje přesnost lokalizace, případně ji zcela znemožňuje. Inerciální navigace je zatížena integrovanými chybami, které velmi zkracují čas, po který lze výstupní data považovat za validní. Popsané způsoby transformace souřadných systémů a navržený způsob detekce segmentů (oblouků) zase vyžadují určité matematické předpoklady pro výpočet, což omezuje jejich aplikaci a nelze je tedy použít za všech situací během jízdy vozidla. Mimo tyto hlavní nedostatky jsou výstupní data jednotlivých subsystémů zatížena dalšími chybami nižších řádů, které dále snižují přesnost.

Jednotlivé subsystémy tedy nelze použít samostatně, ale je nutná jejich kombinace, resp. kombinace - syntéza jejich výstupních dat tak, aby se data vzájemně doplňovala a zpřesňovala. V rámci popisované aplikace pro kolejová vozidla se toto odehrává během zpracování naměřených dat, což bude detailně popsáno v samostatné kapitole. Z hlediska syntézi dat zde dochází k následujícím spojením:

- Projekce GNSS data do mapového podkladu
- Detekce zastávek[3] za pomoci inerciálních dat, GNSS dat a databáze polygonů.
- Indikace rovných úseků a stání za pomoci inerciálních dat a měření času
- Identifikace oblouků a výpočet jejich parametrů z inerciálních dat pomocí algoritmu
- Výpočet délky projetých úseků z inerciálních dat zpřesněný údajem o rychlosti z GNSS přijímače

Takto syntetizovaná data jsou následně prezentována vytvořenou aplikací v textové i grafické podobě, tedy například jako itinerář jízdy nebo poloha promítnutá do mapového podkladu. Příklad těchto výstupů s reálnými daty je zachycen obrázku 3.3. Jedná se o jednu z prvních verzí aplikace, která sloužila především k manuálnímu procházení naměřených dat. Obrázek zde ukazuje průběh zrychlení během jízdy vlaku, na trase Praha-Plzeň. V grafu v levé části je dobře patrný průběh dopředného zrychlení vlaku (červeně), zrychlujícího po průjezdu oblasti s omezenou rychlostí za stanicí Kařez.

[3]Obecně nemusí jít jen o zastávky či stanice, ale o libovolné prostory definované kruhem nebo polygonem na mapě.



Obr. 3.3: Ukázka programu pro syntézu dat z IMU a GPS

3.4 Jiné přístupy

Jak již bylo zmíněno na začátku této kapitoly, problematika určování polohy kolejových vozidel a navigace v dopravě vůbec je řešena prakticky po celém světě. Je proto nutné udržovat povědomí o tom, jaké technologie a principy jsou používány a jaké jsou dosahované výsledky. Z hlediska technického řešení navigační úlohy jsou publikovány stále nové metody a postupy. Některé zajímavé metody a odlišné přístupy jsou uvedeny v následujícím textu.

3.4.1 Implementace více GPS přijímačů a kombinace s DGPS

Navigační systém určený speciálně pro železniční dopravu založený na kombinaci GNSS dat z několika přijímačů GPS, přijímače korekcí DGPS, inerciálních senzorů a výstupu rychloměru, resp. ukazatele ujeté vzdálenosti (odometru), je detailně popsán v [34]. Systém využívá klasickou implementaci Kalmanova filtru, který ve svém stavovém vektoru kombinuje všechny uvedené vstupní informace a dává odhad následujícího stavu (polohy). Tento navigační systém dosahuje velmi dobrých výsledků, ale je celkově komplexnější a vyžaduje hlubší integraci do vozidla.

3.4.2 Detekce výhybek pomocí senzorů vířivých proudů

Tato metoda publikovaná v [35] využívá měření vířivých proudů, tzv. Eddy Current Sensor System (ECS), pomocí něhož je možné detekovat nehomogenitu blízkého feromagnetického materiálu. Princip nasazení na železnici spočívá v montáži ECS senzorů na podvozek kolejového vozidla tak, aby byly senzory umístěny přímo nad kolejnicí (kolejnicemi). Pokud jsou pak ECS senzory dostatečně odstíněné z horní strany, je možné detekovat veškeré změny (přerušení) v kolejivu během jízdy. Tato metoda umožňuje nejen detekci výhybek, ale dokonce i jejich přesnou klasifikaci. Detailní popis ECS systému a dosažené výsledky při detekování výhybek jsou uvedeny v [35].

3.4.3 Určování polohy vlaku na základě měření intenzity rádiového signálu

Tato metoda využívající bezdrátové senzory byla publikována v [36] a pro výpočet polohy vlaku využívá měření intenzity signálu, Received Signal Strength Indicator (RSSI). Pro dosažení požadované přesnosti jsou pak získaná data zpracována metodou sekvenční Monte Carlo. [36]

4

Zařízení pro měření a sběr dat

Aby bylo možné v práci pokračovat a otestovat navržené algoritmy pro detekování charakteristických segmentů, bylo nutné nejprve navrhnout zařízení pro měření a sběr dat.

4.1 Návrh prvního prototypu

První verze měřicího systému byla navržena a realizována v druhé polovině roku 2015 a systém byl složen z počítače (notebooku) ke kterému byly přes USB připojeny komerční GNSS přijímač a speciálně navržená jednotka inerciálního měření. Navigační data i data z inerciálních senzorů byla během měření společně ukládána na disk počítače tak, aby je bylo možné následně načíst a programově zpracovat.

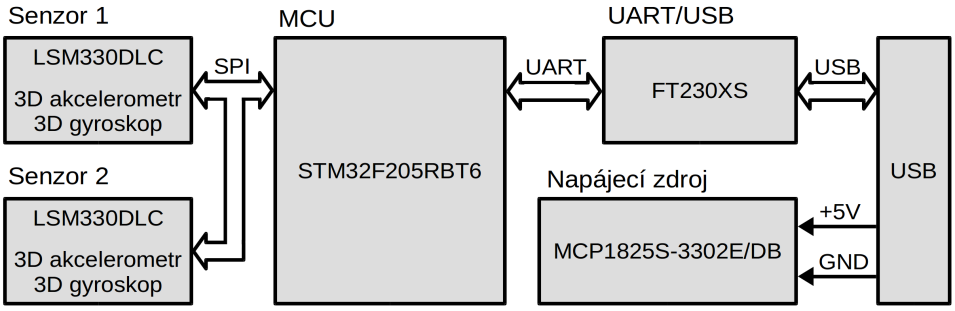
4.1.1 Jednotka inerciálního měření

Navržená jednotka inerciálního měření (IMU) je založená na inerciálních senzorech technologie MEMS. Senzory jsou připojeny k mikrokontroléru, který provádí sběr a filtraci dat a následně data předává dále do připojeného počítače.

Hardwarové uspořádání IMU je dobře patrné z blokového schéma na obrázku 4.1. Mezi klíčové hardwarové součásti zařízení patří především vlastní senzory. Jsou jimi inerciální moduly LSM330DLC, které jsou osazeny v počtu dvou kusů a za pomoci společné SPI sběrnice jsou spojeny s mikrokontrolérem. Ten představuje další důležitou součást a umožňuje získání naměřených dat ze senzorů, jejich zpracování a předání dále do počítače. MCU byl zvolen typu STM32F205RBT6, architektury ARM. Na plošném spoji přípravku jsou dále osazeny stabilizátor napětí a převodník UART na USB, který podporuje USB2.0 Full Speed komunikaci. [37]

4.1.1.1 Hlavní komponenty

Klíčovou součástí zařízení jsou inerciální senzory LSM330DLC, které jsou osazeny v počtu dvou kusů a které v sobě kombinují 3D akcelerometr a 3D gyroskop. Vzhledem k tomu, že tento obvod byl použit i v další verzi měřicího systému, bude podrobněji popsán později.



Obr. 4.1: Blokové schéma první verze navržené IMU

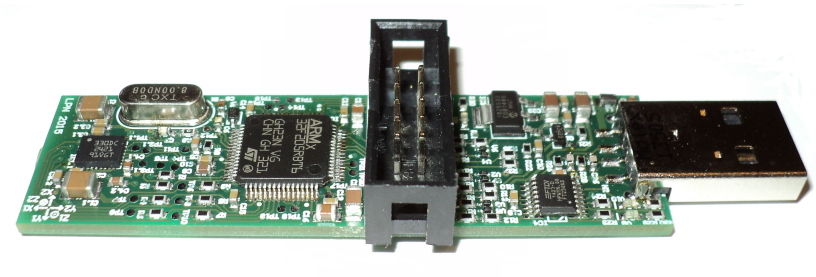
Mikrokontrolér STM32F205RBT6 je 32 bitový, z produkce společnosti STMicroelectronics. Je založen na jádře ARM Cortex M3, disponuje 128 kB Flash paměti a 128 kB SRAM paměti a je proveden v pouzdře LQFP (10 x 10 mm) s 64 vývody. Jedná se o univerzální MCU s bohatou výbavou periferií. Volba tohoto mikrokontroléru byla podpořena především předchozí zkušeností, ale díky kompatibilitě v rámci celé procesorové řady je možné nahradit ho modernějšími obvody, například z řady STM32F400, a to bez vlivu na návrh plošného spoje a s minimálním zásahem do firmwaru. [38]

Z důvodu úspory výpočetního výkonu MCU a také množství práce na firmwaru, nebyla použita USB periferie kterou je mikrokontrolér vybaven, ale byl použit obvod FT230XS, který s procesorem komunikuje prostřednictvím sériové linky UART a celou USB část řeší sám. Na straně PC byl pak použit režim emulace sériového portu, který je kompatibilní s drivery VISA [39], které používá prostředí LabVIEW [40].

4.1.1.2 Mechanické provedení

Jednotka je realizována na jedné dvouvrstvé DPS, o rozměrech 20 x 70 mm, což ji činí dostatečně kompaktní pro přímé připojení do počítače. Na jedné straně DPS je proto osazen USB konektor velikosti A, který umožňuje přímé připojení ke stolnímu počítači či k notebooku. Z USB konektoru je rovněž zajištěno napájení celého zařízení.

Jak bylo zmíněno výše, inerciální moduly LSM330DLC jsou na přípravku osazeny dva, z každé strany DPS jeden. Jejich vzájemná poloha je zvolena tak, že jednotlivé osy, ve kterých probíhá měření fyzikálních veličin, jsou orientovány rovnoběžně a v opačném smyslu. Toto uspořádání přináší další možnosti pro algoritmy, které budou zpracovávat naměřená data a určovat z nich rychlost, polohu, či jiné inerciální veličiny. Provedení celé IMU je dobře vidět na fotografii na obrázku 4.2.



Obr. 4.2: Fotografie první verze IMU

4.1.2 GNSS přijímač

Pro první testování byl zvolen levný GPS/GLONASS přijímač z produkce společnosti Canmore, konkrétně model Canmore GT-730F(G). Přijímač má rozměry asi 70 x 27 x 10 mm a připojuje se přímo do USB konektoru počítače. Komunikace s počítačem probíhá přes sériovou linku převedenou na USB standardním protokolem NMEA 0183, díky čemuž bylo možné přijímač snadno integrovat do LabVIEW aplikace pro sběr dat. Některé parametry přijímače jsou uvedeny v tabulce 4.1.

Tab. 4.1: Vybrané parametry GNSS přijímače Canmore GT-730F(G) [41]

Parametr	**Hodnota**
Chipset	STMicroelectronics (34 kanálů)
Podpora DGPS	SAGPS a SBUS (WAAS a EGNOS)
Protokol	NMEA 0183 v3.01
Připojení k PC	Sériová linka přes USB, 38400 Bd, 8-N-13
Aktualizace výstupu	1 s
Přesnost	Maximálně 2,5 m

4.1.3 Softwarová část

Práci na softwaru lze rozdělit na dvě hlavní části. První z nich je firmware pro mikrokontrolér v rámci výše popsané IMU a druhá zahrnuje aplikace pro PC. Zatímco firmware byl psán klasicky v jazyku C, měřicí a data zpracovávající aplikace pro PC byly vytvořeny v prostředí LabVIEW. Díky grafické povaze programování v LabVIEW, usnadnila tato volba jak tvorbu GUI a reprezentaci výsledků, tak i základní operace jako je čtení sériové linky či zápis dat do souboru. Aplikační část pro PC lze rozdělit na programy pro sběr a ukládání dat a programy pro jejich zpracování, tzv. post-processing. Tyto programy byly vytvořeny v několika verzích tak, jak se postupně vyvíjel celý prototyp. Protože zpracování naměřených dat realizuje hlavní část úlohy určování polohy kolejových vozidel, bude této oblasti později věnována samostatná kapitola.

4.1.4 Firmware pro mikrokontrolér

Z popisu navrženého zařízení je zřejmé, že jeho hlavním úkolem je měření inerciálních veličin a odesílání naměřených dat do připojeného počítače. Této činnosti je tedy podřízen celý návrh firmwaru a mimo vlastní čtení ze senzorů a odesílání dat do PC, umožňuje program ještě například číslicovou filtraci naměřených hodnot, přijímání konfiguračních příkazů z počítače nebo diagnostiku a indikaci provozních stavů.

Program je standardním způsobem členěn do modulů a prakticky vše je psáno v jazyku C[1]. Pro zvolený procesor existuje GCC překladač, takže byl pro překlad zdrojových kódů používán. Programování čipu a ladění programu je realizováno standardním rozhraním J-TAG a hardwarovými prostředky j-link od společnosti Segger.

Jak již bylo uvedeno, hlavním úkolem firmwaru je komunikace se senzory a přenos dat do počítače, a proto jsou tyto části detailněji popsány v následujícím textu.

4.1.4.1 Komunikace se senzory

Z blokového schéma na obrázku 4.1 je zřejmé, že pro komunikaci se senzory bylo zvoleno rozhraní SPI, pro které výrobce uvádí maximální kmitočet hodinových pulzů 10 MHz. Rozhraní pracuje v módu 3, tedy hodinový signál (CLK) je v klidovém stavu v log. 1 a vzorkování dat probíhá na druhou (náběžnou) hranu. Každý přenos obsahuje 16 hodinových pulzů nebo 16 plus násobky 8 při blokovém zápisu či čtení dat. Při čtení je z master zařízení (MCU) odeslán 8 bitový příkaz, na který je v následujících hodinových pulzech odeslána 8 bitová odpověď ze slave zařízení (senzor). Zápis probíhá obdobně, jen po odeslání příkazu následují zapisovaná data odesílaná z MCU. [42]

Výše zmíněný 8 bitový příkaz obsahuje na pozici MSB R/W bit, následovaný MS bitem a zbylých šest bitů obsahuje adresu cílového registru. Jak bývá obvyklé, R/W bit má vždy hodnotu log. 1 pro čtení a hodnotu log. 0 v případě zápisu. MS bit určuje inkrementaci adresy tak, že pokud je jeho hodnota log. 1, je adresa automaticky inkrementována (slouží při blokovém čtení nebo zápisu). SPI sběrnice je společná pro oba senzory, ale ve skutečnosti jde o komunikaci se čtyřmi nezávislými zařízeními[2], z nichž každé má vlastní signál Chip Select (CS) a adresy registrů akcelerometru a gyroskopu se tak mohou překrývat. [42]

Z hlediska softwaru byl pro SPI komunikaci vytvořen modul v jazyku C, který využívá funkce z SPL knihovny. Vzhledem k tomu, že je třeba prostřednictvím SPI obsloužit celkem čtyři zařízení, a to při vyšší komunikační rychlosti (kmitočet hodinových pulzů byl zvolen 5 MHz), byl SPI driver programován s využitím DMA přenosů. To umožňuje efektivně využít SPI sběrnici bez zbytečných prodlev a bez výrazné zátěže CPU. Vlastní ovládání senzorů zajišťuje další modul ve vyšší programové vrstvě, který obsahuje definice všech registrů, parametrů a masek a poskytuje funkce pro čtení měřených dat a nasta-

[1] Několik částí je psáno také v assembleru, například tzv. startup kód.

[2] Každý senzor LSM330DLC obsahuje dva nezávislé čipy (akcelerometr a gyroskop).

vování parametrů senzorů. Modul dále zpracovává externí přerušení od senzorů, které je možné nastavit na spouštění při překročení nastavené prahové hodnoty nebo při zaplnění výstupní FIFO paměti senzoru.

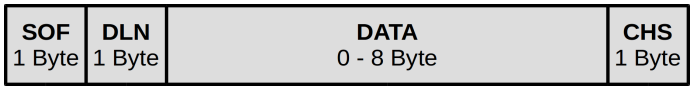
4.1.4.2 Komunikace s počítačem

Komunikace mezi IMU a počítačem je po hardwarové stránce řešena pomocí sériového rozhraní UART, které je převedeno na USB a na straně PC se chová jako virtuální COM port. Komunikace je obousměrná a zahrnuje přenos naměřených, případně i filtrovaných dat ze strany IMU do počítače a naopak, přenos řídicích příkazů ze strany PC do IMU. Ačkoli je komunikace binární, byly, z důvodu možnosti ovládat IMU třeba jen z terminálu, příkazy implementovány tak, že jsou složeny z ASCII znaků. Pro představu jsou některé implementované příkazy uvedeny v tabulce 4.2.

Tab. 4.2: Přehled příkazů pro ovládání první verze IMU

ASCII znaky	**Popis příkazu**
Systémové příkazy	
`**S1T1`	Spustit stream dat přes UART
`**S1T0`	Zastavit stream dat přes UART
`**S1F1` až `3`	Aktivovat příslušný softwarový filtr
`**S1F0`	Deaktivovat všechny softwarové filtry
Příkazy akcelerometrů (kde `x` je `1` nebo `2`)	
`**AxO1` až `9`	Zapnout měření při různých ODR
`**AxO0`	Vypnout měření (výstup) senzoru
`**AxF1` až `4`	Zapnout HP filtr s různými řeznými kmitočty
`**AxF0`	Vypnout HP filtr
`**AxD0`	Manuálně přečíst data (smaže přerušení)
Příkazy gyroskopů (kde `x` je `1` nebo `2`)	
`**GxO1` až `9`	Zapnout měření při různých ODR a BW
`**GxO0`	Vypnout měření (výstup) senzoru
`**GxF1` až `9`	Zapnout HP filtr s různými řeznými kmitočty
`**GxF0`	Vypnout HP filtr
`**GxD0`	Manuálně přečíst data (smaže přerušení)

Pro komunikaci byl také vytvořen jednoduchý datový rámec, s proměnnou délkou, který je ukázán na obrázku 4.3. Každý rámec začíná jedním definovaným znakem, tzv. Start of Frame (SOF), jehož detekce zahajuje příjem rámce. Další bajt, Data Length (DLN) obsahuje informaci o délce přenášených dat, která může být v rozsahu 0 až 8 bajtů. Následující přenášené bajty jsou datové a celý rámec je zakončen jedním bajtem kontrolního součtu, Checksum (CHS). Ten je dán prostým součtem všech předchozích bajtů rámce včetně SOF na šířce 8 bitů.



Obr. 4.3: Datový rámec komunikace mezi první verzí IMU a PC

Takto realizovaná komunikace mezi IMU a PC je maximálně jednoduchá, relativně rychlá a poskytuje i minimální formu zabezpečení. Implementace na obou komunikačních stranách pak vede na jednoduchý stavový automat. Maximální přenosová rychlost je omezena převodníkem UART/USB na 3 MBd [37] a ovladačem VISA na straně LabVIEW na 921600 Bd [39]. Tím byla tedy definována přenosová rychlost na 921600 Bd.

4.2 Druhá verze zařízení pro měření a sběr dat

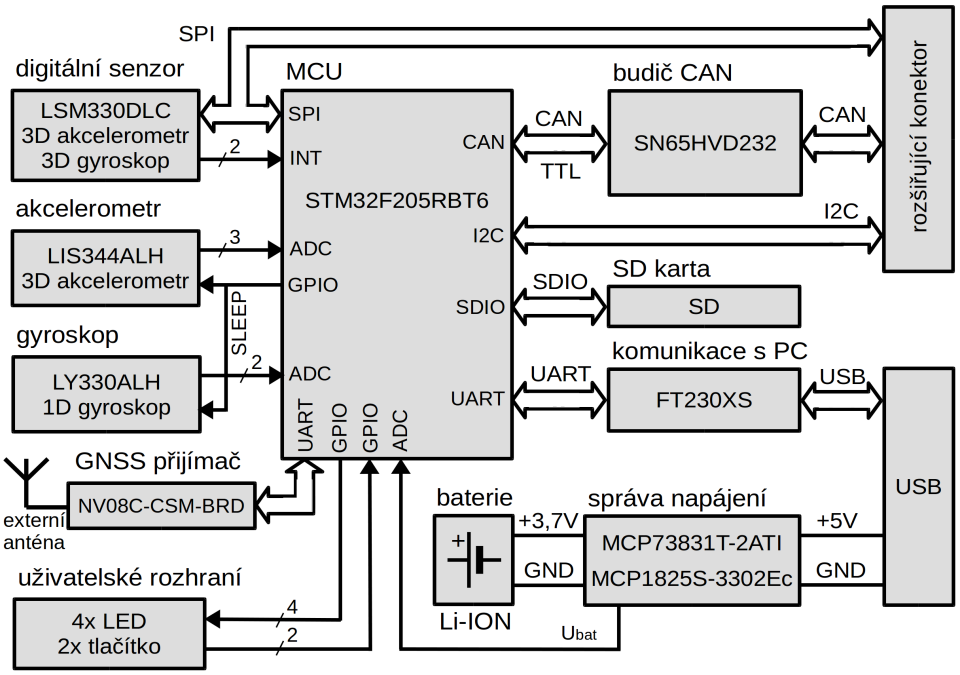
Druhá verze zařízení pro inerciální měření a sběr navigačních dat vznikla v první polovině roku 2016. Na základě zkušeností z předchozí verze zde byla provedena řada úprav tak, aby zařízení bylo kompaktnější, přesnější a mohlo pracovat samostatně. Jak IMU část, tak i GNSS přijímač byly integrovány do jednoho robustního kovového pouzdra. GNSS přijímač byl vybaven aktivní externí anténou a napájení celého systému zajišťuje jeden článek Li-ION baterie. Pro funkci zařízení není tedy nadále třeba připojovat počítač a naměřená data jsou namísto odesílání do počítače ukládána na paměťovou kartu.

4.2.1 Koncepce a hlavní hardwarové součásti

Uspořádání všech částí zařízení je dobře patrné z blokového schéma na obrázku 4.4. Jádro IMU části je tvořeno senzorickými obvody tříosého akcelerometru LIS344ALH, jednoosého gyroskopu LY330ALH a senzorem LSM330DLC. První dva jmenované obvody mají analogový výstup a jsou tedy připojeny k ADC vstupům mikrokontroléru STM32F205RBT6. Třetí senzor integruje tříosý akcelerometr a tříosý gyroskop s pokročilými funkcemi a komunikace s ním probíhá pouze číslicově přes sběrnici SPI. Část GNSS je tvořena modulem NV08C-CSM z produkce švýcarské společnosti NVS Technologies [43]. Přijímač je připojen k procesoru sériovým rozhraním UART a komunikace probíhá prostřednictvím protokolu NMEA.

Vzhledem k tomu, že použitý mikrokontrolér disponuje rozhraním pro paměťové karty SDIO, je SD karta pro záznam naměřených dat připojena plnohodnotně, nikoli pouze pomocí sběrnice SPI. Napájení celého zařízení zajišťuje baterie Li-ION, která je dobíjena vždy, když je zařízení připojeno k počítači přes USB. Získání dostatečného proudu z počítače zajišťuje obvod FT230XS, který o dostatečný proud požádá během enumerace v rámci USB protokolu. Vlastní regulaci proudu a napětí během nabíjení řeší nabíjecí obvod MCP73831T, který je přímo určený pro správu článků typu Li-ION o nominálním napětí 3,7 V. [44] Pro konfigurování zařízení, ladění programu a přenos souborů z paměťové karty je využíváno sériové rozhraní, které výše zmíněný obvod FT230XS převádí na

USB. Systém pak lze ovládat z počítače textově, pomocí sériového terminálu nebo za tím účelem vytvořenou aplikací.



Obr. 4.4: Blokové schéma druhé verze měřicího zařízení

Jako rozhraní pro ovládání zařízení bez připojeného počítače, slouží dvě tlačítka a čtyři LED diody. Tlačítky lze například spouštět a zastavovat měření, restartovat procesor nebo zahájit přenos souboru. Pomocí LED jsou pak indikovány různé provozní stavy či chyby a například také to, zda GNSS přijímač zachytil dostatečný počet družic a navigační data jsou tedy validní.

Poslední hardwarovou součástí, která stojí za zmínku je budič sběrnice CAN, který může být v případě potřeby na desku dodatečně osazen. Důvodem této přípravy je možnost zapojit měřicí zařízení do palubního systému vozidla a získat tak přístup k dalším údajům, jako je například údaj z rychloměru.

4.2.1.1 Inerciální senzory

Stejně jako v první verzi měřicího zařízení, i zde byl použit digitální senzor LSM330DLC z produkce společnosti STMicroelectronics označovaný obchodním názvem iNEMO. Jedná se o integrovaný obvod, ve kterém jsou pomocí technologie MEMS realizovány tříosý akcelerometrický senzor a tříosý gyroskopický senzor. Obvod pak dále obsahuje číslicovou část, která zajišťuje vzorkování výstupního napětí senzorů, registry pro nastavení prahových hodnot na jejichž základě lze generovat přerušení, SPI a I2C komunikační rozhraní, výstupní FIFO paměť a další součásti. Číslicová část také umožňuje konfigurovat několik integrovaných filtrů typu dolní a horní propust. Protože je obvod primárně určen pro spotřební elektroniku a mobilní zařízení, obsahuje další pokročilé funkce, jako například detekce pohybu, poklepání či detekce volného pádu. [42]

Při výběru obvodu LSM330DLC byla důležitá také jeho dostupnost. Obvody tohoto typu jsou často vyrobeny ve velké sérii pro určitého výrobce mobilních zařízení a následně se také objeví na trhu, ale v podstatě už se nevyrábí a dostupnost je tak nejistá. Další důležitá vlastnost je integrace jak 3D akcelerometru, tak i gyroskopu. Některé pokročilejší moduly v sobě ještě integrují například kompas a magnetometr a jsou tak prodávány jako kompletní IMU řešení. Tyto moduly však často rozměrově přesahují velikost běžného integrovaného obvodu a podléhají přísnější legislativě. Navíc, vzhledem k charakteru měření a prostředí ve kterém je zařízení instalováno, není použití magnetometru a kompasu příliš vhodné. Tyto senzory fungují spolehlivě spíše při umístění ve větší výšce a větší vzdálenosti od kovové konstrukce vozidla a využívají se například v navigačních systémech UAV, jako jsou různé drony apod. Z těchto důvodů bylo od samého začátku práce na tématu určování polohy kolejových vozidel počítáno pouze s použitím inerciálních senzorů typu akcelerometr a gyroskop.

V popisovaném měřicím zařízení je obvod iNEMO využíván pouze v režimu kontinuálního měření zrychlení a úhlové rychlosti a kromě filtrů nejsou další funkce využívány. Z důvodu rychlosti přenosu je pro komunikaci se senzorem využíváno rozhraní SPI s frekvencí hodinových pulzů 5 MHz a dva signály přerušení, které indikují procesoru, že senzor má připravená nová data. Parametry senzorů tohoto typu jsou napříč součástkovou základnou velmi podobné a jsou dány vlastní technologií MEMS, jejíž přínos není ve vyšší přesnosti ale spíše v možnosti integrace do malého prostoru a ve výrazně nižší ceně. [12] Hlavní parametry obvodu LSM330DLC jsou uvedeny v tabulce 4.3.

Tab. 4.3: Vybrané parametry obvodu LSM330DLC [42]

Parametr	**Hodnota**
Rozsah akcelerometru	Volitelný: ±2 g, ±4 g, ±8 g a ±16 g
Rozsah gyroskopu	Volitelný: $\pm 250^{\circ}\cdot s^{-1}$, $\pm 500^{\circ}\cdot s^{-1}$ a $\pm 2000^{\circ}\cdot s^{-1}$
Citlivost akcelerometru	V závislosti na rozsahu: 0,012 g až 0,001 g
Citlivost gyroskopu	V závislosti na rozsahu: $(7\cdot 10^{-2})^{\circ}\cdot s^{-1}$ až $(8{,}75\cdot 10^{-3})^{\circ}\cdot s^{-1}$
ODR akcelerometru	Volitelný: 1 Hz až 1,344 kHz
ODR gyroskopu	Volitelný: 95 Hz až 760 Hz
Šířka pásma gyroskopu	Volitelná: 12,5 Hz až 100 Hz
Šum akcelerometru	Při rozsahu ±2 g a ODR 1,344 kHz: $220\,\mu g/\sqrt{Hz}$
Šum gyroskopu	Při rozsahu $\pm 250^{\circ}\cdot s^{-1}$ a šířce pásma 50 Hz: $0{,}03^{\circ}\cdot s^{-1}/\sqrt{Hz}$

Pro možnost porovnání a zpřesnění měřených inerciálních veličin byla druhá verze měřicího systému doplněna o další dva senzory, tentokrát s analogovým napěťovým výstupem. Jedná se o tříosý akcelerometr LIS344ALH a jednoosý gyroskop LY330ALH. Oba obvody jsou opět z produkce společnosti STMicroelectronics a jsou provedeny technologií MEMS v pouzdrech LGA. [45], [46]

Akcelerometry tohoto typu jsou běžně dostupné s podobnými parametry a jako al-

ternativa byly také vyzkoušeny obvody z produkce společnosti NXP[3]. U gyroskopických senzorů s analogovým výstupem je dostupnost na trhu menší a v úvahu přichází prakticky pouze obvody od společností STMicroelectronics nebo Analog Devices. Tyto součástky také podléhají přísnější legislativě a do některých regionů je distributoři nedodávají[4]. Klíčové parametry zvolených obvodů jsou uvedeny v tabulce 4.4.

Tab. 4.4: Vybrané parametry obvodů LIS344ALH a LY330ALH [45], [46]

Parametr	**Hodnota**
Rozsah akcelerometru	Volitelný: $\pm 2\,\mathrm{g}$ a $\pm 6\,\mathrm{g}$
Rozsah gyroskopu	$\pm 300°\cdot s^{-1}$
Citlivost akcelerometru	Při rozsahu $\pm 2\,\mathrm{g}$: $\frac{Ucc}{5}\,\mathrm{V/g}$
	Při rozsahu $\pm 6\,\mathrm{g}$: $\frac{Ucc}{15}\,\mathrm{V/g}$
Citlivost gyroskopu	$3{,}752\,\mathrm{mV}/°\cdot s^{-1}$
Nelinearita akcelerometru	$\pm 0{,}5\%$ z rozsahu
Nelinearita gyroskopu	$\pm 1\%$ z rozsahu
Šířka pásma gyroskopu	$140\,\mathrm{Hz}$
Šum akcelerometru	Při rozsahu $\pm 2\,\mathrm{g}$: $50\,\mu\mathrm{g}/\sqrt{\mathrm{Hz}}$
Šum gyroskopu	$0{,}014\,°\cdot s^{-1}/\sqrt{\mathrm{Hz}}$

Další funkcí popsaných obvodů je tzv. režim spánku (sleep mode), během něhož je výrazně snížena spotřeba obvodu z jednotek miliampér na jednotky mikroampér. [45], [46] Vzhledem k tomu, že měřicí systém je bateriově napájený, je tato funkce aktivována vždy, když právě neprobíhá měření.

4.2.1.2 GNSS modul

Přijímač GNSS byl použit ve verzi NV08C-CSM-BRD, což je vlastně OEM deska o rozměru 50 x 35 mm na které je osazen vlastní modul. Na desce je také vyřešena vysokofrekvenční část, takže lze snadno připojit externí anténu pomocí koaxiálního konektoru MCX. Konektory pro připojení antény jsou zde dva, jeden pro pasivní a druhý pro aktivní anténu. [47] Pro ilustraci je na obrázku 4.5 uvedena fotografie modulu.

Pro propojení desky s procesorem je použit konektor s počtem pinů 20 a roztečí 2 mm. Konektor je osazen na stejné straně jako anténní konektory, což popisované aplikaci nevyhovovalo a bylo nutné konektor osadit na druhou stranu tak, aby bylo možné GNSS modul umístit na základní desku popisovaného měřicího systému a konektory MCX přitom směřovaly vzhůru. Propojovací konektor obsahuje mimo napájecí vodiče a některé další speciální signály také dvě sériová rozhraní typu UART označená jako A a B. Na rozhraní A lze s modulem komunikovat protokolem NMEA 0183, verze 2.3, který již byl

[3]Dříve Freescale.

[4]ČR do těchto regionů ve většině případů také spadá.

detailně popsán výše. Rozhraní UART B pak používá proprietární protokol BINR, který je, na rozdíl od textového NMEA, binární a funguje na principu dotaz-odpověď. [47]

Obr. 4.5: Fotografie GNSS modulu NV08C-CSM-BRD

Závěrem popisu GNSS přijímače NV08C je nutné říci, že jde o špičkové zařízení, které splňuje průmyslové standardy a požadavky a výsledky dosažené s tímto přijímačem jsou nesrovnatelné oproti přijímači použitému v první verzi popisovaného systému pro určování polohy. Pro přesnější představu jsou v tabulce 4.5 uvedeny vybrané parametry přijímače.

Tab. 4.5: Vybrané parametry GNSS modulu NV08C-CSM-BRD [47]

Parametr	**Hodnota**
Napájecí napětí	3,3 V
Proudový odběr	60 mA až 90 mA
Provozní teplota	-40°C až 80°C
GNSS systémy	GPS, GLONASS, GALILEO
Počet kanálů	32
Studený start	Průměrně 25 s
Přesnost (chyba) polohy	Průměrně menší než 1,5 m
Přesnost (chyba) výšky	Průměrně menší než 2 m
Přesnost (chyba) rychlosti	Průměrně menší než 0,05 $m \cdot s^{-1}$
Obnovovací frekvence	Volitelně: 1, 2, 5 a 10 Hz
Limity	Rychlost menší než 500 $m \cdot s^{-1}$
	Zrychlení menší než 5 g
	Nadmořská výška menší než 50000 m
Komunikační protokoly	NMEA 0183 / IEC 61162-1
	BINR (binární)
	RTCM SC-104
Konfigurace UART	Pro NMEA: 1N8, 115200 Bd

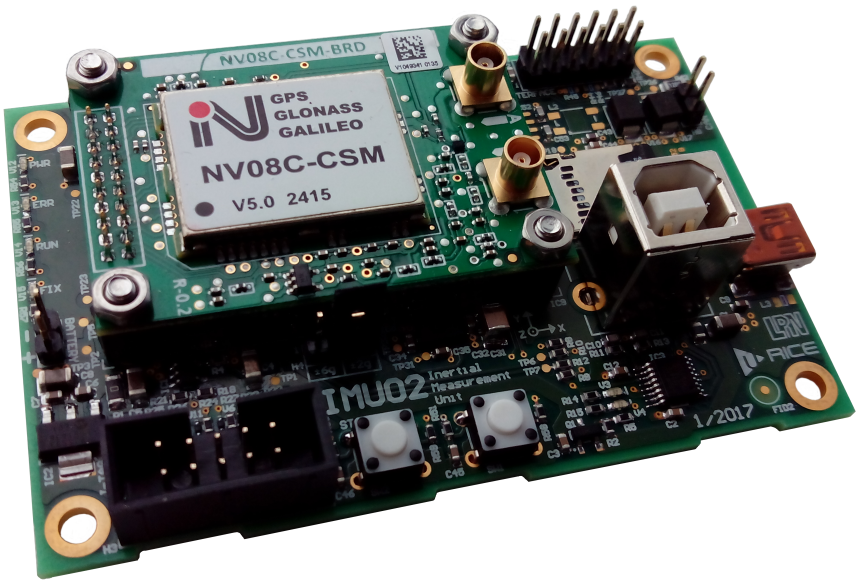
V popisu GNSS přijímače výše, byla zmíněna také externí anténa. Ta byla použita aktivní, napájená napětím 3,3 V přímo z modulu NV08C. Anténa je značky PCTEL, označení 8117D a je přímo doporučena výrobcem přijímače pro použití s moduly NV08C. Je určena pro příjem družicových signálů GPS L1 a GLONASS a její frekvenční rozsah je

1568 až 1618 MHz. Při napájení 3,3 V a odběru 9 mA je zesílení nízkošumového zesilovače asi 28 dB a šum 1,5 dB. Anténa je určena pro průmyslové použití s teplotním rozsahem -40°C až 85°C a krytím IP67. Ve čtvercové základně antény o rozměru asi 45 x 45 mm je umístěn magnet, takže anténu je možné snadno připevnit k ocelovým konstrukcím a povrchům. [48]

Proudový odběr modulu NV08C se pohybuje v rozmezí 60 až 90 mA plus odběr aktivní antény. Z tohoto důvodu je využita možnost uvést modul GNSS do režimu spánku vždy, když právě neprobíhá měření. Přechod do tohoto režimu se provádí softwarově, odesláním příslušné věty v rámci protokolu NMEA. [47]

4.2.2 Mechanické provedení

Zařízení je realizováno na jedné desce plošných spojů o rozměru 85 x 56 mm na které jsou osazeny senzory, mikrokontrolér, slot pro SD kartu, FTDI čip, nabíjecí obvod a všechny ostatní podpůrné obvody a komponenty. K této desce je pomocí čtyř distančních sloupků délky 6 mm a příslušného konektoru připojen také GNSS modul. Pro připojení k počítači jsou na desce volitelně osaditelné dva typy USB konektorů, jeden MiniUSB a druhý typu B. Konektor USB B je vertikální a lze ho používat v případě, že je deska namontována v pouzdru zařízení a MiniUSB na hraně desky tak nelze použít. Pro programování mikrokontroléru a případné ladění programu je na desce osazen standardní 10 pinový J-TAG konektor s roztečí 2,54 mm, který sice zabírá více místa, ale výrazně snižuje nároky na hardwarové nástroje pro programování. Na desce je umístěn ještě jeden konektor s roztečí 2,54 mm, který na svých 12 pinech zpřístupňuje sběrnice SPI, I2C a CAN, jakožto možné rozšíření zařízení o další funkce. Pro přesnější představu je na obrázku 4.6 uvedena fotografie sestavy popsané desky plošných spojů s instalovaným GNSS modulem.



Obr. 4.6: Fotografie sestavy DPS měřicího systému s osazeným GNSS přijímačem

Aby bylo možné nechat zařízení měřit data ve vozidle samostatně po dobu několika hodin, bylo zařízení navrženo jako bateriově napájené a umístěno do kompaktního uzavřeného pouzdra. Celek je vidět na obrázku 4.7. Na dně krabičky z hliníkové slitiny je fixována baterie a nad ní je přišroubovaná sestava DPS. Napětí z baterie je k desce připojeno běžným konektorem s roztečí 2,54 mm. Anténní konektor GNSS přijímače typu MCX je za pomoci úhlového protikusu a 50 mm dlouhého koaxiálního kabelu typu RG178, vyveden na vnější stranu krabičky, na koaxiální konektor typu SMA. Díky tomuto řešení, je možné po zapojení baterie a aktivaci měření tlačítkem, celou krabici uzavřít víčkem a zajistit šrouby. Zařízení je tak chráněno proti různým vnějším vlivům a zásahům, což je velmi praktické a měření probíhá zcela nerušeně.

Obr. 4.7: Fotografie navrženého zařízení pro měření a sběr dat

Ve spodní části krabičky jsou dvě křídla s otvory, což umožňuje snadnou montáž zařízení. Vzhledem k tomu, že ve vozidlech ve kterých probíhá měření, není zpravidla možné provádět jakékoli úpravy za účelem montáže, byly na spodní část pouzdra přišroubovány čtyři magnety, které jsou dostatečně silné k tomu, aby zařízení bylo po celou dobu měření pevně fixováno na ocelových částech konstrukce vozidla. Například bylo možné zařízení umístit na boční přístrojový panel v kabině tramvaje.

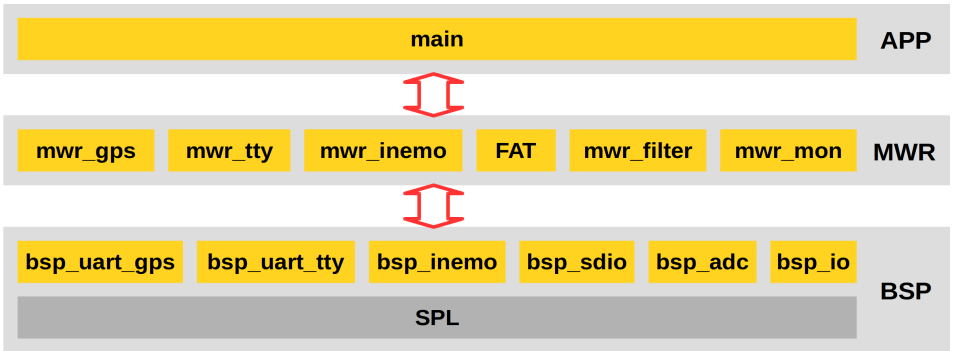
4.2.3 Firmware pro mikrokontrolér

Stejně jako v případě první verze zařízení, tak i v druhé verzi byl firmware programován v jazyku C za použití stejných nástrojů a SPL knihoven od STMicroelectronics. Mimo již popsanou komunikaci s digitálním senzorem LSM330DLC je zde navíc implementováno

čtení analogových hodnot ze senzorů LIS344ALH a LY330ALH, k čemuž je využívána ADC periferie mikrokontroléru. Další změnou je ukládání naměřených a filtrovaných hodnot na SD kartu, narozdíl od přímého odesílání do počítače, jako tomu bylo v předchozí verzi. Pro komunikaci s SD kartou je využívána SDIO periferie MCU a díky implementaci systému FAT, jsou data na kartu ukládána ve standardizovaném formátu, který je čitelný v běžných počítačích. Implementace FAT systému byla zvolena ze zdroje [49], která je pojata minimalisticky a je tedy vhodná pro integraci do firmwaru jednočipových počítačů.

Ačkoli je zařízení primárně ovládáno manuálně tlačítky, byla zachována sériová linka pro komunikaci s počítačem a zařízení je možné z připojeného počítače také ovládat. Pro zjednodušení, je komunikace řešena čistě textově, prostřednictvím terminálu. Výjimkou je pouze režim přenosu souborů, kdy je ze zařízení přenášen vybraný soubor, uložený na SD kartě, v binárním režimu. Nově implementovanou částí je také komunikace s GNSS modulem, která probíhá v rámci protokolu NMEA po sériovém rozhraní UART.

Pro konkrétní představu o rozsahu programování, je na obrázku 4.8 uvedena struktura celého popisovaného firmwaru. Zdrojové kódy tvoří jednotlivé moduly, které lze zařadit do vrstev podobně, jak je to definováno v ISO/OSI modelu. Nejnižší vrstva, která je pevně svázaná s použitým hardwarem je zde nazvána BSP a obsahuje také SPL knihovnu. Nad SPL knihovnou se nachází ovladače periferií (`bsp_sdio`, `bsp_adc`, `bsp_io`) a linkové vrstvy komunikačních protokolů (`bsp_uart_gps`, `bsp_uart_tty`, `bsp_inemo`). Další vrstva označená MWR pak obsahuje platformě nezávislé moduly, jako jsou vlastní komunikační protokoly (`mwr_gps`, `mwr_tty`, `mwr_inemo`), systém FAT, modul číslicových filtrů `mwr_filter` a modul `mwr_mon`, který slouží k monitorování uživatelských vstupů a některých veličin. Nejvyšší vrstva uvedená na schématu je aplikační (označeno APP) a obsahuje v tomto případě pouze modul `main`. Zde je realizována vlastní funkce zařízení a z hlavní programové smyčky jsou volány funkce nižších vrstev.



Obr. 4.8: Struktura firmwaru druhé verze měřicího zařízení

Kromě SPL knihovny, která je dodávána výrobcem procesoru[5] a systému FAT, který byl použit ze zdroje [49], byly všechny moduly navrženy a naprogramovány v rámci práce

[5]Standard Peripheral Library tvoří HAL vrstvu nad registry mikrokontroléru. [38]

na popisovaném projektu. V následujícím textu jsou vytvořené zdrojové kódy stručně popsány, zejména z hlediska jejich funkce.

4.2.3.1 Komunikace s digitálním senzorem

Komunikace se senzorem LSM330DLC a s jeho dvěma částmi je řešena v modulech `bsp_inemo` a `mwr_inemo`. Modul `bsp_inemo` obsahuje inicializaci rozhraní SPI a s tím souvisejících přerušení a DMA kanálů. Exportovány jsou funkce pro inicializaci SPI a DMA, včetně zadání pointerů na datové buffery, funkce pro rekonfiguraci DMA přenosu a funkce pro povolení a zakázání DMA přenosu. Vlastní nastavení SPI již bylo podrobně popsáno výše a další popis tedy není nutný.

Modul `mwr_inemo` realizuje vlastní komunikační protokol a umožňuje inicializaci a konfiguraci senzoru[6], čtení a zápis do jeho registrů a čtení naměřených dat. Některé funkce jsou blokující, ale většina protokolu je realizována stavovým automatem jako neblokující. Exportovány jsou funkce pro inicializaci modulu, inicializaci senzoru, zápis a čtení registrů, čtení dat, reset senzoru, povolení přerušení od senzoru, funkce vracející stav komunikace, funkce vracející poslední naměřená data atd. Důležitá je také procesní funkce stavového automatu, která musí být pravidelně volána z hlavní programové smyčky a dvě callback funkce pro přerušení od SPI přenosu. Z digitálního senzoru jsou čteny 16 bitové znaménkové číselné hodnoty, tedy v rozsahu -32768 až 32767. Číslo je úměrné měřené veličině působící ve směru dané osy, přičemž nulová hodnota odpovídá stavu, kdy žádná veličina nepůsobí. V rámci akcelerometru jsou čteny tři hodnoty pro tři osy a stejně tomu je i u gyroskopického senzoru. Celkem je tedy čteno 12 bajtů dat. Spouštění čtení je odvozeno od externích přerušení, které jsou generovány senzorem a perioda čtení pak odpovídá nastavené hodnotě ODR. U akcelerometrické části senzoru je hodnota ODR nastavena na 100 Hz a u gyroskopu je to 95 Hz. Data nejsou filtrována, ale pouze průměrována, neboť filtrace je řešena přímo v čipu senzoru, viz. popis senzoru výše a zdroj [42].

4.2.3.2 Čtení hodnot z analogových senzorů

Vzorkování napěťových výstupů z analogových senzorů zajišťuje ADC periferie mikrokontroléru, která je inicializována v modulu `bsp_adc`. Zde je nastaven také DMA kanál, kterým jsou navzorkované hodnoty ukládány na určené místo v paměti a časovač, od kterého je odvozeno spouštění ADC převodu. Mimo inicializační funkci modul dále exportuje procesní funkci, která je volána periodicky, z hlavní smyčky programu a která ukládá nově navzorkované hodnoty do bufferu a volá nad nimi filtrační funkce. Poslední dvě funkce pak slouží ke zjištění zda jsou k dispozici nová filtrovaná data a k jejich případnému přečtení.

Vzorkováno je celkem sedm hodnot, z nichž pět souvisí přímo se senzory a jsou to osy x, y a z od akcelerometru a výstupní a referenční hodnota od gyroskopu. Zbylé dvě veličiny jsou pak napětí baterie[7] a výstup vnitřního teplotního senzoru, kterým je

[6] Obě části senzoru (akcelerometr a gyroskop) se konfigurují samostatně.

[7] Systém je bateriově napájený a je tedy vhodné mít k dispozici informaci o aktuálním stavu baterie.

mikrokontrolér vybaven. Perioda vzorkování ADC je nastavena na 100 μs a číslicový filtr vrací filtrovanou hodnotu při každém 128 volání. Z toho plyne, že nová filtrovaná data jsou k dispozici každých 12,8 ms, což odpovídá frekvenci přibližně 78 Hz.

Všechny navzorkované, filtrované či přečtené hodnoty ze senzorů, analogových i digitálních, jsou úměrné měřené veličině, ale jsou bez fyzikálního rozměru. Přepočet na fyzikální veličiny není ve firmwaru realizován, protože je to výpočetně náročné[8] a vzhledem k pozdějšímu zpracování dat na počítači, také neúčelné. Přepočet je řešen až při zpracování naměřených dat prostřednictvím programů vytvořených v LabVIEW. Výjimkou jsou pouze napětí baterie a teplotní senzor. Ty jsou přepočítávány na reálné fyzikální veličiny tak, aby byly k dispozici již v průběhu měření a program v MCU mohl adekvátně reagovat, například ukončit měření a uzavřít soubor na SD kartě při nízkém napětí baterie tak, aby nedošlo k jeho poškození pokud by systém, vlivem nízkého napětí, přestal náhle zcela fungovat. Přepočet je realizován v modulu `mwr_mon`, který za tímto účelem exportuje dvě funkce. Tyto funkce vrací hodnotu napětí baterie v mV a teplotu čipu ve °C.

K analogovým senzorům lze ještě přiřadit několik funkcí z modulu `bsp_io`, které ovládají příslušný pin mikrokontroléru tak, aby bylo možné obvody LIS344ALH a LY330ALH uvádět do režimu spánku a zpět, jak to bylo popsáno výše.

4.2.3.3 Filtrace měřených veličin

Dalším modulem, který je použit při čtení analogových senzorů, je modul filtrace `mwr_filtr`. Tento modul exportuje funkci realizující polyfázový číslicový filtr, popsaný v kapitole 2.2.2.2. Funkce je napsána tak, že filtrace probíhá současně nad pěti hodnotami a všechny koeficienty filtru jsou vlastně poli. Realizace číslicového filtru v MCU bez speciálních hardwarových prostředků je vždy náročná na množství potřebné paměti a v tomto konkrétním případě je pro filtraci využíváno cca 300 bajtů RAM paměti. Další funkce exportované modulem jsou funkce pro průměrování typu MAV, které jsou volány například nad vzorkovanými hodnotami napětí baterie a teplotního senzoru.

4.2.3.4 Komunikace s GNSS modulem

Komunikace mezi MCU a GNSS přijímačem je řešena v modulech `bsp_uart_gps` a `mwr_gps` a jde o běžný způsob implementace sériové komunikace rozhraním UART. Hardwarové nastavení periferie UART, které již bylo částečně popsáno v kapitole o GNSS přijímači, je realizováno v modulu `bsp_uart_gps` exportovanou inicializační funkcí. Ta nastavuje UART na přenosovou rychlost 115200 Bd a ostatní parametry tak, jak je to specifikováno výrobcem přijímače v tabulce 4.5. Periferie je obsluhovaná v přerušení, DMA režim není vzhledem k množství přenášených dat nutný. Další exportované funkce modulu

[8] Zpravidla se přepočet neobejde bez použití datového typu float, neboť na celočíselném rozsahu dochází k značným ztrátám přesnosti, zvláště pak při dělení.

jsou funkce pro odeslání a příjem jednoho bajtu dat, které nepracují přímo s registry periferie, ale s buffery definovanými v modulu. Tyto buffery jsou nastaveny na délku 256 bajtů pro příjem a 32 bajtů pro vysílání, což s rezervou postačuje pro bezpečné přijetí celé NMEA věty. Poslední dvě funkce modulu jsou určeny pro obsluhu přerušení a realizují vlastní odesílání a příjem.

Protokol NMEA, který byl podrobně popsán v kapitole 2.1.4, je implementován v modulu `mwr_gps` a jeho periodicky volaná procesní funkce obsahuje stavový automat, který zajišťuje příjem a zpracování definovaných vět. Pokud je přijatá věta typu RMC (volitelně i GGA), je uložena jako řetězec do bufferu, odkud je pomocí funkce předávána jako pointer dál, například funkci zapisující data na SD kartu. Další funkce exportovaná modulem umožňuje uvést GNSS přijímač do režimu spánku, a to pomocí příslušné NMEA věty, která je do přijímače poslána. Poslední funkce, která stojí za zmínku, provádí jednoduché parsování přijaté věty a vrací informaci o tom, zda jsou GNSS data platná, tedy přijímač je ve stavu tzv. fixed.

4.2.3.5 Ukládání dat na SD kartu

Implementovaný FAT systém je sám o sobě složen z několika modulů psaných v jazyku C a byl převzat z [49], jako volně šiřitelný zdrojový kód pod licencí typu GNU GPL. Je určen pro embedded systémy a je psán velmi minimalisticky a úsporně a nevyžaduje OS nebo jiný scheduler založený na vláknech. Je pouze nutné napsat nižší softwarovou vrstvu pro práci s připojeným diskovým zařízením, v tomto případě SDIO driver pro SD kartu. Na druhou stranu, takto pojatá implementace FAT systému, tedy bez RTOS, vede na blokující chování volaných funkcí. Tento fakt je třeba brát v úvahu a vhodně ošetřit časově kritické operace. Subsystém FAT exportuje řadu funkcí pro práci se soubory a souborovým systémem, například funkce připojení a odpojení svazku, pro otevření a zavření souboru, synchronizaci a smazání, funkce pro čtení a zápis, pro procházení souborem a nastavování ukazatele atd. [49]

Modul `bsp_sdio` řeší vlastní komunikaci s SD kartou a je to nadstavba SPL knihovny pro SDIO periferii. Modul obsahuje celou řadu funkcí pro inicializaci periferií jako je SDIO a DMA, funkce pro inicializaci SD karty, čtení stavu karty, blokový a multi-blokový zápis a čtení, funkce pro mazání atd. Modul také obsahuje definice příkazů, které jsou pro SD kartu definovány ve specifikaci [50].

Vlastní ukládání dat je řešeno v aplikační vrstvě, tedy v modulu `main`. Zde se nachází periodicky volaný úsek kódu, který v případě, že je aktivní měření, zapisuje do nově otevřeného CSV souboru data v textové formě. Soubor je tedy snadno čitelný (nejen strojově) a plně přenositelný. Perioda zápisu dat je nastavena na 100 ms. Pro představu, jaká data jsou během měření ukládána, je v tabulce 4.6 uvedena hlavička ukládaného CSV souboru.

Tab. 4.6: Hlavička CSV souboru ukládaného během měření

Položka	**Popis**
TIME	Časová známka - hodnota milisekundového čítače
GPRMC	RMC věta - celý řetězec tak, jak byl přijat z přijímače
ACCX	Analogový akcelerometr osa X - navzorkovaná a filtrovaná hodnota
ACCY	Analogový akcelerometr osa Y - navzorkovaná a filtrovaná hodnota
ACCZ	Analogový akcelerometr osa Z - navzorkovaná a filtrovaná hodnota
GREF	Analogový gyroskop reference - navzorkovaná a filtrovaná hodnota
GOUT	Analogový gyroskop výstup - navzorkovaná a filtrovaná hodnota
VBAT	Napětí baterie - navzorkovaná a průměrovaná hodnota
TEMP	Teplota čipu - navzorkovaná a průměrovaná hodnota
DAX	Digitální akcelerometr osa X - 16 bitové neznaménkové číslo
DAY	Digitální akcelerometr osa Y - 16 bitové neznaménkové číslo
DAZ	Digitální akcelerometr osa Z - 16 bitové neznaménkové číslo
DGX	Digitální gyroskop osa X - 16 bitové neznaménkové číslo
DGY	Digitální gyroskop osa Y - 16 bitové neznaménkové číslo
DGZ	Digitální gyroskop osa Z - 16 bitové neznaménkové číslo

4.2.3.6 Manuální ovládání zařízení a indikace provozních stavů

Jak bylo popsáno v kapitole zabývající se hardwarovou částí měřicího systému, zařízení je vybaveno dvěma tlačítky a několika LED diodami pro ovládání a indikaci provozních stavů. Jak tlačítka, tak i LED diody jsou ovládány GPIO piny procesoru. Z hlediska firmwaru se jedná o modul `bsp_io`, který exportuje funkce pro nastavení LED a čtení tlačítek. Vyhodnocení stisknutí tlačítka a odstranění zákmitů je realizováno v modulu `mwr_mon`, v příslušné funkci, která je volána periodicky z programové smyčky.

LED diody ovládané procesorem jsou na desce osazeny čtyři a význam jejich indikace je uveden v tabulce 4.7. Tlačítka jsou k dispozici dvě, z nichž jedno je označeno *START/STOP* a slouží ke spouštění a zastavování měření a druhé je označeno *TEST* a má více funkcí současně. Pokud je aktivní nějaký chybový stav indikovaný LED, je stisknutím tlačítka test chyba kvitována. Není-li žádná chyba aktivní a je-li aktivován přenos souboru, je stisknutím tlačítka test zahájen přenos. Poslední funkce tlačítka, to jest softwarový reset MCU, se uplatní pouze pokud je tlačítko stisknuto ve chvíli, kdy není aktivní žádná chyba ani přenos souboru.

Mimo čtyři LED indikátory řízené mikrokontrolérem, jsou na desce ještě další tři. Dvě oranžové jsou osazené poblíž obvodu FT230XS a indikují komunikaci s PC a jedna červená LED je umístěna poblíž nabíjecího obvodu. Tato dioda pak svým svitem indikuje nabíjení baterie.

Tab. 4.7: Indikace provozních stavů zařízení

Název LED	**Barva**	**Indikace**
PWR	Zelená	Svítí - zařízení je v provozu
		Nesvítí - program MCU neběží
ERR	Červená	Svítí - indikace chybového stavu [a)]
		Nesvítí - žádná chyba
RUN	Zelená	Svítí - probíhá měření nebo přenos souboru
		Nesvítí - měření ani přenos neprobíhají
FIX	Zelená	Svítí - GNSS data jsou platná
		Nesvítí - GNSS data platná nejsou

[a)] Další informace o chybě je možné vyčíst pomocí připojeného počítače.

4.2.3.7 Ovládání přes sériovou linku a přenos souborů

Ve druhé verzi měřicího systému slouží sériová komunikace s počítačem pouze k ladění a konfigurování zařízení a během měření se zpravidla nepoužívá. Nebyl zde proto implementován žádný protokol, ale komunikace probíhá pouze textově v terminálu.

Periferie UART je nastavována v modulu `bsp_uart_tty` a vše je prakticky shodné s výše popsaným modulem `bsp_uart_gps`. Rozdíl je pouze ve velikosti bufferů, přijímací má délku 64 bajtů a vysílací 128 bajtů a v přenosové rychlosti, která je po zapnutí nastavena na 9600 Bd. Do modulu byla dále doplněna funkce, kterou lze přenosovou rychlost měnit na 115200 Bd nebo 921600 Bd. Právě hodnota 921600 Bd je využívána pro přenos souborů[9].

Druhý modul, který se podílí na komunikaci s PC je `mwr_tty`. Procesní funkce tohoto modulu zpracovává přijaté znaky a detekuje zda byl přijat platný příkaz. Přijaté příkazy jsou ukládány do bufferu a lze je postupně vyčítat k tomu určenou exportovanou funkcí. Implementované příkazy jsou uvedeny v tabulce 4.8. Modul dále exportuje funkci `write()`, která je volána funkcí `printf()` ze standardní knihovny, takže je výstup směrován na terminál počítače. Tato funkce je využívána zejména pro ladění. Poslední funkce kterou stojí za to uvést, slouží k vkládání příkazů do bufferu. Tímto způsobem lze vyvolat reakci na přijatý příkaz také programově, což zjednodušuje a zpřehledňuje aplikační část firmwaru.

Několikrát byl také zmíněn přenos souboru z SD karty v zařízení do připojeného počítače. Tato funkce je řešena přímo v modulu `main` a byla implementována čistě z praktických důvodů, aby nemusela být SD karta vyjímána po každém měření. Princip přenosu je následující. Příslušným příkazem z terminálu počítače je aktivován režim přenosu souboru, načež se překonfiguruje sériová linka na rychlost 921600 Bd a systém čeká na stisknutí tlačítka test. Po stisknutí pak začne samotný přenos.

[9] Maximální rychlost kterou podporují drivery VISA používané v prostředí LabVIEW. [39]

Tab. 4.8: Přehled příkazů pro ovládání druhé verze IMU

ASCII znaky	**Popis příkazu**
Systémové příkazy	
`:00`	Prázdný příkaz - žádná akce
`:01`	Vrátí status zařízení (také teplotu a napětí baterie)
`:02`	Vrátí seznam aktivních chyb
`:03`	Smaže aktivní chyby [a)]
`:04` až `:06`	Nastaví přenosovou rychlost na 9600, 115200 nebo 921600 Bd
Příkazy pro měření	
`:10`	Aktivuje přesměrování dat z GNSS přijímače do terminálu
`:11`	Deaktivuje přesměrování
`:12`	Zahájí měření (data se ukládají na SD kartu) [b)]
`:13`	Ukončí měření [b)]
Příkazy pro práci se soubory	
`:20`	Vrátí seznam souborů s naměřenými daty na SD kartě
`:30` až `:39`	Přenos souboru 0 až 9 do počítače
`:40` až `:49`	Smazání souboru 0 až 9 z SD karty

[a)] Stejná funkce jako stisknutí tlačítka *TEST*.

[b)] Stejná funkce jako stisknutí tlačítka *START/STOP*.

Byla snaha implementovat standardizovaný protokol Kermit, který je pro přenos souborů určený, respektive jeho modernizované open-source provedení vhodné pro embedded zařízení E-Kermit [51]. Bohužel, implementace protokolu E-Kermit nebyla nakonec možná, z důvodu nedostatku paměti v MCU. Přenos tedy probíhá binárně, bajt po bajtu, bez jakékoli kontroly. Kontrolu je ale možné provést tak, že porovnáme velikost přijatého souboru a velikosti, kterou lze získat příslušným příkazem ze zařízení. Bezpečnější, ale časově náročnější metodou je přijetí souboru dvakrát po sobě a jejich následné porovnání.

5

Zpracování naměřených dat

Z popisu hardwaru v předchozí kapitole plyne, že navržené měřicí zařízení slouží především ke sběru dat z reálného provozu a k jejich ukládání. Z oblasti zpracování dat je zde implementována pouze popsaná číslicová filtrace, případně prosté průměrování několika po sobě jdoucích vzorků. Vlastní zpracování dat je prováděno později, po skončení měření v počítači za pomoci programů vytvořených v prostředí LabVIEW. Z hlediska zpracování dat se tedy jedná o tzv. off-line metodu.

Z tohoto pohledu je také možné rozdělit vytvořené programy na dvě skupiny, z nichž první pracuje nad celým vzorkem dat a druhá zpracovává data vzorek po vzorku, bez znalosti celkového počtu vzorků. První skupina programů byla vytvořena převážně v počáteční fázi celého projektu a slouží především k zobrazování a analýze sebraných dat. Druhá skupina programů pak byla vytvářena od začátku tak, aby byl balík naměřených dat zpracováván výhradně vzorek po vzorku stejně, jako by byla data právě měřena. Toto řešení pak umožňuje vytvořené navigační algoritmy nasadit i v tzv. režimu on-line, což je kýženým cílem popisované problematiky určování polohy kolejových vozidel.

Z hlediska reálné implementace přináší prostředí LabVIEW také možnost exportu vytvořených algoritmů přímo do modulu s FPGA, což může být výhoda. Obecně lze ale říci, že realizovat počítač s dostatečným výpočetním výkonem pro běh LabVIEW v tzv. embedded formě, není v dnešní době zásadní problém a na trhu existuje dostatek hotových řešení. Z tohoto důvodu byly sběr a zpracování dat řešeny od začátku jako dva samostatné celky, jejichž spojení však nevyžaduje příliš úsilí. V krajním případě by například bylo možné pro první realizaci navigačního systému umístit do vozidla mimo měřicí zařízení také notebook s příslušnými programy, napojený do sítě internet, například přes mobilní datovou službu a výstupy navigačního systému tak přenášet pokusně k operátorovi dispečingu nebo na jiná místa.

5.1 Software LabVIEW

Než budou popsány vlastní programy a algoritmy, je zde vhodné uvést několik základních informací o zmíněném softwaru LabVIEW a zdůvodnit jeho použití. Název LabVIEW

je zkratka pro termín Laboratory Virtual Instrument Engineering Workbench a jedná se o softwarové programovací prostředí vyvíjené společností National Instruments (NI) od druhé poloviny osmdesátých let. Práce v tomto prostředí je založena na grafickém programování[1], které zjednodušuje a urychluje práci, takže se člověk může soustředit na řešený problém bez nutné znalosti syntaxe a klíčových slov, jak tomu bývá u konvenčních programovacích jazyků. Software LabVIEW je určen především pro měřicí a testovací aplikace v laboratořích i v průmyslu a jeho hlavní výhodou v této oblasti je snadný přístup k měřicím přístrojům a k hardwaru obecně. To je nejvíce dáno tím, že prostředí LabVIEW je dnes v oblasti měření a sběru dat považováno za jeden ze standardních nástrojů, takže výrobci přístrojů dodávají příslušné knihovny a ovladače. [40]

Další výhodou prostředí LabVIEW je jeho grafický charakter. Každý program[2] obsahuje dvě části, jednu s grafickým zdrojovým kódem, která se nazývá Block Diagram [40] a druhou, která se nazývá Front Panel [40] a obsahuje zobrazovací a ovládací prvky jako jsou tlačítka, textová pole, grafy, číselné indikátory atd. Reprezentace naměřených a zpracovaných dat v grafické podobě je zde tedy automaticky zahrnuta a vytvářena už během programování. Tento fakt opět urychluje práci a prakticky odpadá nutnost programování nějakého speciálního GUI apod.

Ačkoli popisovaná aplikace navigačního systému plně nevyužívá možnosti spojení hardwaru a softwaru, které LabVIEW nabízí, bylo dalším důvodem pro použití tohoto prostředí velké množství knihoven VI, které instalace standardně obsahuje. V tomto konkrétním případě se jedná především o VI z oblasti matematiky (lineární algebra, geometrie), zpracování signálu (filtrace, spektrální analýza), práce se soubory (čtení/zápis CSV), práce s I/O (čtení sériových linek), práce s poli a mnoha dalších. Důležitou vlastností programování v LabVIEW je také možnost vkládat VI z knihoven i vlastní vytvořená do dalších nadřazených VI a tvořit tak stromovou strukturu celého složitějšího programu. Toto zapouzdření je z hlediska přehlednosti pro grafické programování zcela zásadní.

Komplexnost a šíře záběru softwarového balíku LabVIEW je skutečně rozsáhlá a další popis by byl značně nad rámec tohoto textu. Více informací lze případně najít přímo na webových stránkách společnosti NI v [40]. Hlavní důvody pro nasazení LabVIEW v rámci popisované aplikace je možné shrnout v následujících bodech:

- Velké množství připravených VI z oblasti matematiky a zpracování signálu
- Integrované nástroje pro práci se soubory včetně formátu CSV
- Vestavěné ActiveX komponenty (zobrazení mapy pomocí webového prohlížeče)
- Jednoduchá implementace komunikace přes sériový port[3]

[1] Přesný termín je Visual Programming Language (VPL), resp. Dataflow Visual Programming Language (DFVPL). [52]

[2] V prostředí LabVIEW se program nazývá VI podle přípony zdrojového souboru.

[3] Dodávaný balík ovladačů Virtual Instrument Software Architecture (VISA) [39], umožňuje pracovat s různými standardními sběrnicemi jako je sériový port, USB, GPIB apod. [39]

- Snadná a rychlá tvorba GUI pro ovládání programu a zobrazení dat
- Možnost realizovat záznam, zpracování i prezentaci měřených dat jedním nástrojem

5.2 Záznam a analýza dat

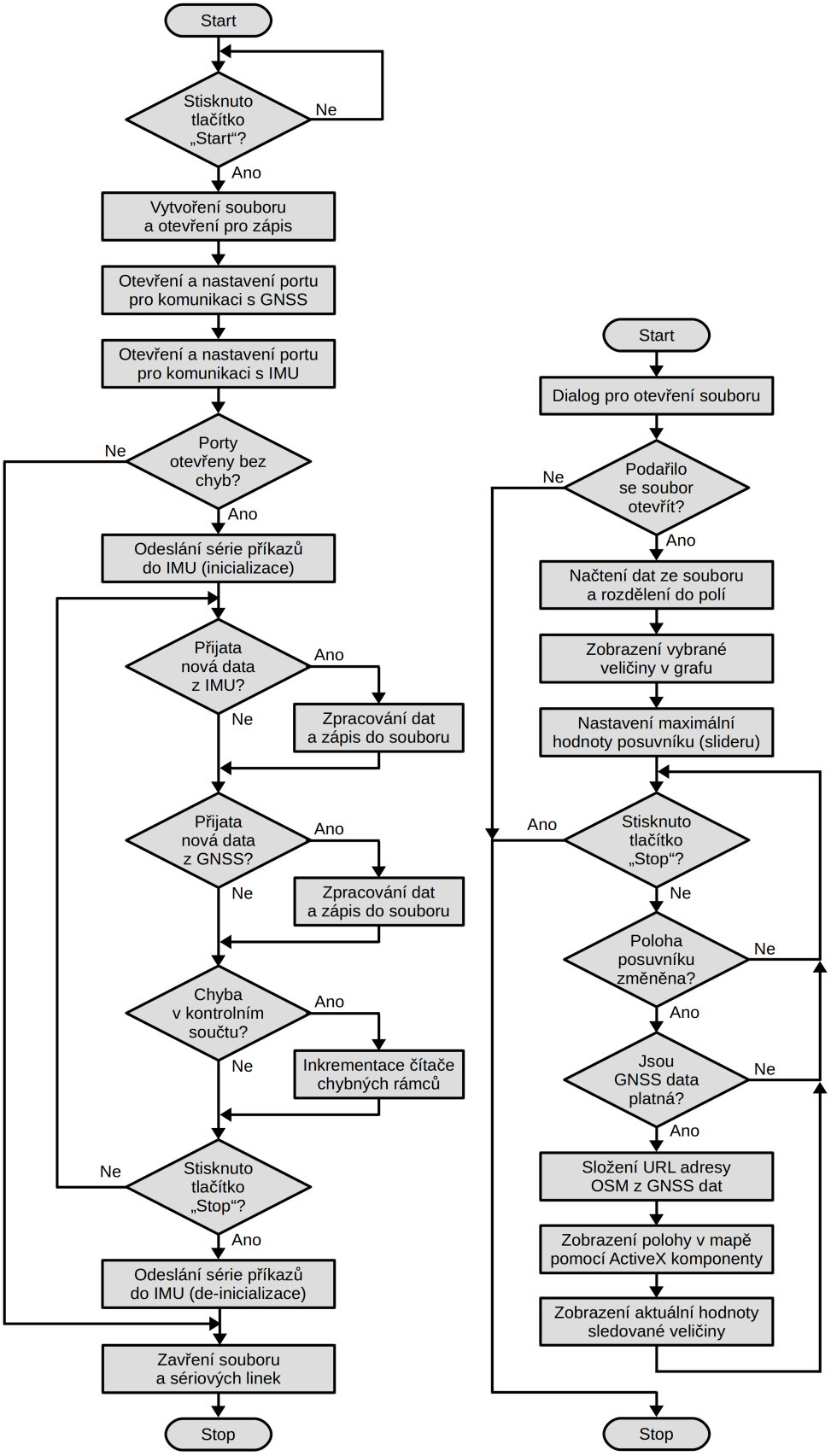
Program vytvořený pro záznam dat úzce souvisí s první verzí měřicího zařízení, která byla popsána v kapitole 4.1. Měření probíhalo tak, že ve vozidle byl umístěn notebook s připojeným zařízením pro sběr dat a GNSS přijímačem a v počítači bylo spuštěné příslušné VI. Program periodicky četl data z přijímače i z měřicího zařízení a ukládal je do souboru na disku.

Následně bylo vytvořeno několik VI pro zobrazení získaných dat a pro realizaci některých výpočtů. Tyto programy umožnili zobrazovat a manuálně procházet zaznamenané nebo vypočítané průběhy, případně automaticky procházet pole dat a hledat v nich určité artefakty.

5.2.1 Program pro záznam dat

Funkce programu pro záznam dat spočívá ve čtení dat z IMU jednotky a GNSS přijímače a jejich ukládání do souboru na disku počítače. Popis programu z hlediska jeho funkce je lépe realizovatelný pomocí vývojového diagramu, který je uveden na obrázku 5.1 a). Po spuštění hlavního VI, čeká program na stisknutí tlačítka *Start*, které je zobrazeno na panelu VI. Po stisknutí tlačítka volá program další VI[4], která vytvoří a otevře nový soubor pro zápis dat a inicializují obě sériové linky pro komunikaci s připojenými zařízeními. Pokud vše proběhne bez chyby (zařízení jsou připojené apod.), jsou do IMU odeslány příkazy (viz. tabulka 4.2), které nastaví parametry měření a zahájí přenos dat. Od této chvíle pak běží program ve smyčce s periodou 20 ms, přičemž dostatečná přesnost časování je zajištěna použitím struktury *Timed Loop*. Ve smyčce jsou kontrolovány vstupní buffery sériových linek a případně zpracovávána přijatá data, která jsou zapisována do souboru. Také jsou kontrolovány chyby v komunikaci (kontrolní součty) a případně inkrementovány čítače chyb rámce, jejichž hodnoty jsou zobrazeny na čelním panelu. Celý proces je pak ukončen po stisknutí tlačítka *Stop*, po němž následuje odeslání série příkazů do jednotky IMU, uzavření souboru a sériových linek a ukončení programu.

[4]V terminologii LabVIEW jsou podprogramy, tedy VI volaná z jiného VI, nazývány sub-VI. [40]



(a) Program pro záznam dat (b) Program pro procházení dat

Obr. 5.1: Vývojové diagramy pomocných programů

V rámci programu pro záznam dat bylo vytvořeno několik VI, resp. sub-VI, které řeší především komunikaci se zařízeními a zpracování přijatých (textových) dat. Tato VI jsou uvedena v tabulce 5.1. Za bližší zmínku stojí dvě z uvedených VI, a to *GpsGPRMC.vi* a *Parser.vi*. První uvedené VI zpracovává přijaté bajty z GNSS přijímače ve kterých hledá RMC větu, kterou následně za pomoci sub-VI s názvem *GpsToken.vi* parsuje a data ukládá do struktury[5]. Druhé uvedené VI, *Parser.vi*, obsahuje stavový automat, jež zpracovává bajty přijaté z IMU a skládá z nich datové rámce, které validuje pomocí kontrolního součtu. Výstupem je pak textový řetězec obsahující všechna přijatá inerciální data, připravený pro zápis do souboru a také logická hodnota indikující případnou chybu v kontrolním součtu.

Tab. 5.1: Přehled VI vytvořených v rámci programu pro záznam dat

Název VI	**Funkce**
AppGrab05.vi	Hlavní VI programu pro záznam dat
GpsGPRMC.vi	Čtení a parsování RMC věty přijaté z GNSS přijímače
GpsToDeg.vi	Přepočet souřadnic z formátu NMEA na stupně
GpsToken.vi	Vrací část RMC věty (token) na základě delimiteru
IncrementIfTrue.vi	Na základě logického vstupu inkrementuje vstupní hodnotu
Parser.vi	Stavový automat pro zpracování datových rámců z IMU
GpsToString.vi	Převod vybraných GNSS dat na textový řetězec

Jak bylo uvedeno výše, program pro záznam dat byl vytvořen pro první prototyp měřicího zařízení, kde splnil svůj účel a v současné době již není jako celek využíván. Některé části, jako například zpracování NMEA vět, ale bylo možné, díky jejich univerzálnosti, použít i pro další práci a mírně upravená VI jsou využívána i v programu pro zpracování dat, který bude popsán později.

5.2.2 Procházení inerciálních a GNSS dat

Programy pro procházení dat byly tvořeny jak pro první verzi měřicího systému, tak i později pro systém s druhou verzí IMU jednotky. Rozdíl mezi verzemi programů většinou spočívá pouze v provedení sub-VI, které načítá a zpracovává soubor s naměřenými daty. Druhá verze měřicího systému totiž zapisuje do souboru více dat.

Na obrázku 5.1 b) je znázorněn vývojový diagram jednoho z programů pro procházení dat. Po spuštění VI je pomocí standardního dialogu vybrán soubor s daty, který je následně otevřen a načten. Data jsou rozdělena z CSV dle sloupců, převedena zpravidla na číslo typu double a uspořádána do polí. Vybraná veličina nebo i více veličin je zobrazeno v grafu a na základě délky pole (počtu záznamů v souboru) je nastavena maximální hodnota ovládacího prvku typu posuvník (slider). Následující část programu pak běží ve smyčce

[5]V prostředí LabVIEW se objekt typu struktura nazývá cluster.

s periodou přibližně 100 ms a reaguje buď na stisknutí tlačítka *Stop* nebo na posunutí slideru. Při posunutí slideru je jeho hodnota použita k indexování pole GNSS dat i pole hodnot sledované veličiny. Pokud jsou aktuální GNSS data platná, je z nich vytvořena URL adresa na mapový server OSM a pomocí integrovaného ActiveX prvku webového prohlížeče je zobrazena mapa s vyznačenou polohou. Zobrazena je též hodnota sledované veličiny pro aktuální polohu navigovaného objektu.

Společným cílem programů pro procházení dat byla zpravidla možnost spojení dat GNSS, resp. polohy vozidla na mapě s průběhy měřených veličin v dané lokaci. Tímto způsobem je pak možné najít na mapě průjezd určitým segmentem tratě a pozorovat jak se v daném úseku chovají inerciální veličiny. Na základě získaných poznatků pak byly navrhovány navigační algoritmy a výpočetní zpracování inerciálních veličin.

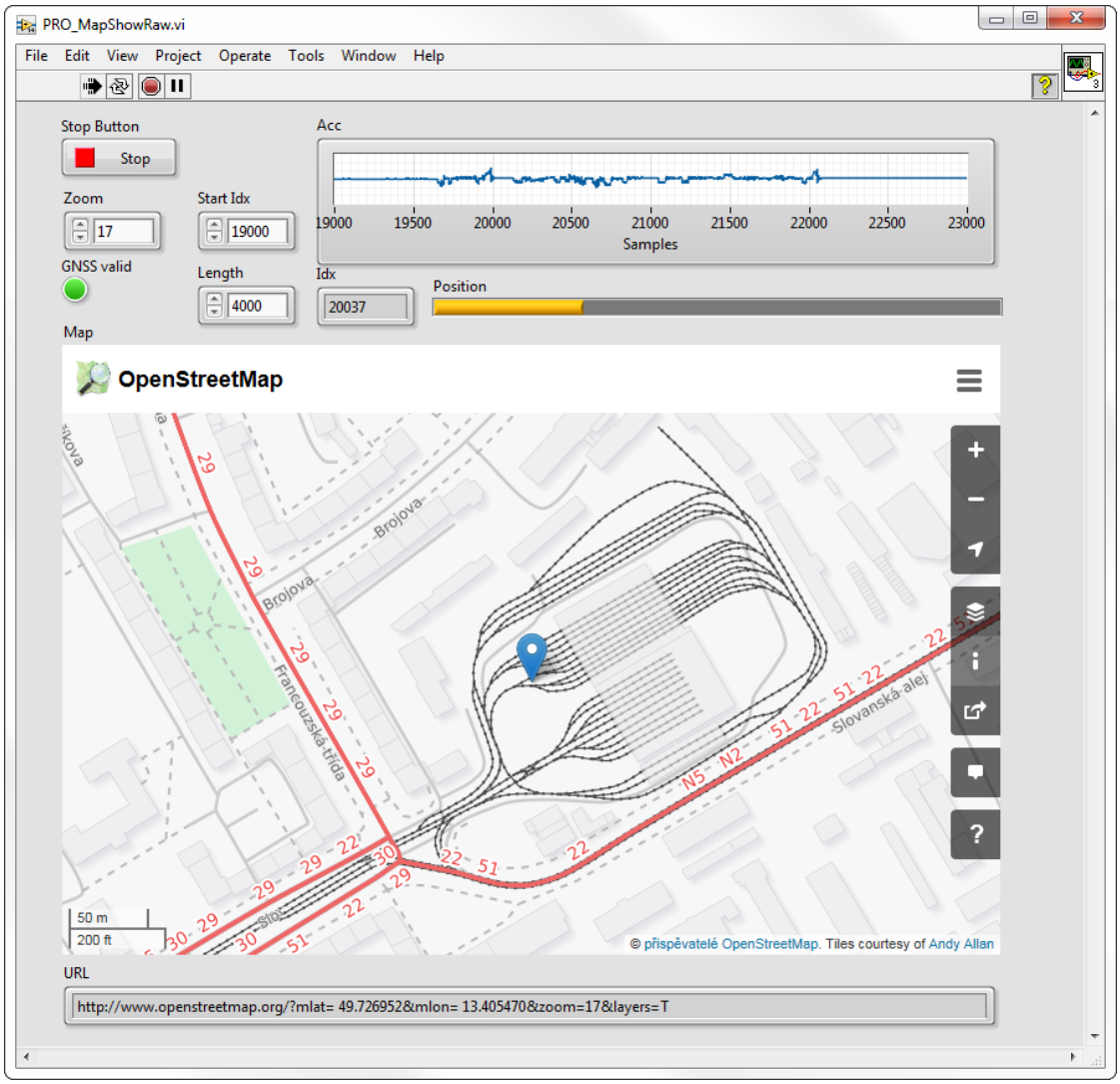
Pro konkrétní představu o práci s popsaným programem je na obrázku 5.2 ukázán čelní panel VI za běhu. Zde se jedná o procházení dat naměřených v tramvajovém provozu, konkrétně jde o tramvajovou linku č. 2 v Plzni. Jedna ze starších verzí programu již také byla uvedena na obrázku 3.3 v kapitole 3.3. Tam se jednalo o data z železniční dopravy.

5.2.3 Další podpůrné programy

Tato část programování v LabVIEW zahrnuje vytváření různých VI ve kterých byly postupně aplikovány a zkoušeny matematické výpočty uváděné v teoretické části, ale které nakonec nebyly pro finální aplikaci použity nebo byly použity jen některé části.

Jiným podpůrným programem je VI pro přenos souborů s naměřenými daty z měřicího zařízení do počítače tak, jak to bylo popsáno v kapitole 4.2.3.7. Po spuštění tohoto VI je otevřen sériový port pro komunikaci se zařízením a nový soubor pro zápis dat. Pokud vše proběhne správně, odešle program do zařízení příkaz pro zjištění seznamu měřicích souborů uložených na SD kartě a vyčká na odpověď. Přijatý seznam je následně zobrazen a program čeká na stisknutí tlačítka *Continue*. V tuto chvíli je také možné zvolit číslo souboru který se má přenášet. Po stisknutí tlačítka odešle program příkaz pro přenos příslušného souboru, následovaný příkazem pro nastavení maximální komunikační rychlosti[6]. Program ještě počká na dokončení komunikace, zavře port a znovu ho otevře s již nastavenou vyšší rychlostí. VI pak běží ve smyčce a každý přijatý bajt zapisuje do souboru na disku tak dlouho, dokud není stisknuto tlačítko *Stop*. Vlastní přenos je indikován neustále aktualizovaným počtem přenesených bajtů, který je zobrazován na panelu VI. Pokud je tlačítko stisknuto, je odeslán příkaz pro nastavení původní komunikační rychlosti, sériová linka i soubor jsou uzavřeny a program skončí.

[6]Maximální rychlost je 921600 Bd, viz. kapitola 4.2.3.7.



Obr. 5.2: Ukázka programu pro procházení dat

5.3 Hlavní navigační program

Hlavní navigační program je v podstatě hledaným výstupem celé práce, resp. té části, která se zabývá zpracováním dat a programováním v LabVIEW. Program sestává z více než dvou desítek VI, ve kterých jsou algoritmizovány a implementovány matematické výpočty popsané v kapitolách 2 a 3. Jsou zde zpracovávány jak inerciální veličiny z IMU, tak i údaje o poloze z GNSS a systém také pracuje se souborem obsahujícím databázi definovaných prostorů. Tento fakt naplňuje zmiňovaný princip syntézy dat z několika zdrojů. Důležitou vlastností je také způsob zpracování měřených dat vzorek po vzorku, což je nutný předpoklad pro on-line implementaci programu.

Navigační program bude v této kapitole detailně popsán, včetně některých důležitých sub-VI, ze kterých je složen. Protože se později v textu vyskytnou názvy jednotlivých VI, je pro přehled uveden v tabulce 5.2 seznam všech VI, vytvořených v rámci programu.

Tab. 5.2: Přehled VI vytvořených v rámci hlavního navigačního programu

Název VI	**Funkce**
Test07Stable.vi	Hlavní VI navigačního programu
AccGetDt.vi	Výpočet časového intervalu mezi dvěma po sobě jdoucími vzorky
AccToPitchAndRoll.vi	Výpočet úhlů klonění (roll) a klopení (pitch) a transformační matice
CalculateBeta.vi	Výpočet úhlu natočení v rámci algoritmu pro výpočet poloměru oblouku
CalculateRadius.vi	Výpočet poloměru projetého oblouku
IntegrateAngle.vi	Výpočet úhlu natočení integrací úhlové rychlosti
IntegrateDistance.vi	Výpočet ujeté vzdálenosti integrací rychlosti
IntegrateVelocity.vi	Výpočet rychlosti integrací zrychlení
Map2DTo1DStringArray.vi	Pomocná funkce pro čtení souboru s definovanými prostory
MapPointDetection.vi	Detekce zda navigovaný objekt vstoupil/opustil/je v prostoru definovaném kruhem
MapPointDetectionPolygon.vi	Detekce zda navigovaný objekt vstoupil/opustil/je v prostoru definovaném polygonem
MapShow.vi	Zobrazení navigovaného objektu na mapě
MiscEdgeDetector.vi	Detekce vzestupné a/nebo sestupné hrany v posloupnosti vzorků
MiscIntegrate.vi	Průběžně integruje vstupní veličinu
PeakAngleDetection.vi	Detekování a výpočet dosaženého úhlu během jednoho souvislého otáčení
PeakDistanceDetection.vi	Výpočet souvisle ujeté vzdálenosti
ReadDataFile.vi	Načtení souboru s naměřenými daty a rozdělení do polí vzorků

Tabulka pokračuje na další straně . . .

Tab. 5.2 – pokračování

Název VI	**Funkce**
VehicleToGravityTransformation.vi	Přepočet vektoru zrychlení do soustavy orientované ve směru tíhového zrychlení
VehicleToHeadingTransformation.vi	Přepočet vektoru zrychlení do soustavy orientované ve směru jízdy vozidla
AverageCounter.vi	Průměruje tři vstupní hodnoty po zadaný čas
DirectionCosines.vi	Implementace metody směrových kosinů pro strap-down algoritmus
ElementRotMatrix.vi	Vrací elementární rotační matici na základě zadaného úhlu a osy
Transform.vi	Provede maticové násobení vstupního vektoru a transformační matice
TresholdBand.vi	Hodnoty vstupních signálů menší než nastavený práh nahradí nulou

5.3.1 Popis funkce programu

Po startu hlavního VI začíná program načtením souboru s naměřenými daty. Na základě nastavení přepínače na hlavním panelu, je načten buď soubor dle cesty zadané v příslušném poli nebo je vyvolán standardní dialog pro výběr souboru. Vlastní načtení pak provádí sub-VI s názvem *ReadDataFile.vi*, které vrátí několik polí obsahujících jednotlivé měřené veličiny. Vzhledem k tomu, že soubor s daty může být relativně dlouhý, řádově stovky tisíc záznamů, jsou pole dat před zpracováváním vzorek po vzorku oříznuta, dle zadaného počátečního indexu (čísla vzorku) a délky (počtu vzorků). Tato funkce je pouze pomocná a slouží k omezení délky trasy, například pokud je zkoumán jen jeden určitý úsek mezi dvěma zastávkami nebo pokud byla v rámci jednoho měření stejná trať projížděna vícekrát. Pro on-line realizaci pak tato funkce již nemá smysl.

5.3.1.1 Inerciální výpočty

Následující část programu již pracuje s jednotlivými vzorky dat, resp. indexuje postupně zmíněná pole. Pomocí dvou prvků typu Numeric Control na ovládacím panelu, resp. pomocí jejich nastavených hodnot, je několik prvních vzorků zahozeno a několik dalších je pouze průměrováno. Toto řešení jednak eliminuje chybná data, která vznikla při zapínání měřicího přístroje ve vozidle (otřesy způsobené stisknutím tlačítka, náběh číslicového filtru a MAV filtrů apod.), zároveň připraví stabilní hodnoty určitých veličin pro výpočet počátečních podmínek. Zde je důležité zmínit, že předpokladem pro správnou funkci navigačního systému je stacionární stav navigovaného objektu v počátku měření. To v praxi znamená, že aktivace měření může proběhnout pouze ve chvíli kdy vozidlo stojí a kromě

tíhového zrychlení na něj nepůsobí žádná další inerciální veličina. Mezi určované počáteční podmínky patří transformační matice pro přepočet vektoru zrychlení do soustavy orientované ve směru tíhového zrychlení. Tento výpočet provádí VI s názvem *AccToPitchAndRoll.vi*, které na výstupu vrací úhly klopení (pitch) a klonění (roll) a vlastní transformační matici. Další průměrovanou veličinou je úhlová rychlost bočení (yaw), která je měřena analogovým gyroskopem. Její hodnota zjištěná v této fázi je použita jako počáteční kompenzace offsetu tohoto senzoru.

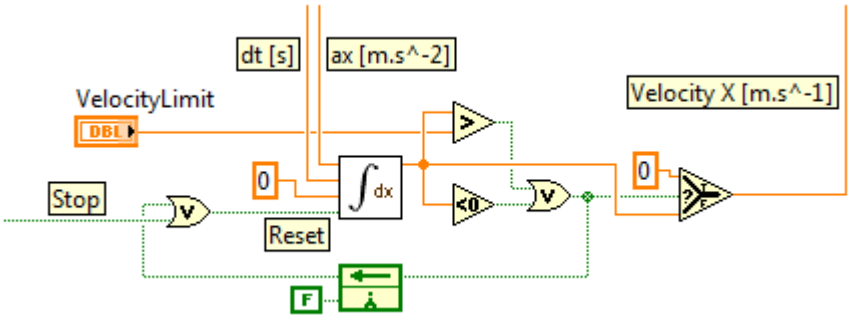
Dále program běží ve smyčce a postupně iteruje zbývající části polí a provádí nad nimi výpočty. Jednou z veličin ukládaných při měření je také čas, resp. časová značka od milisekundového čítače. Sub-vi s názvem *AccGetDt.vi* z této hodnoty, resp. ze dvou po sobě jdoucích hodnot, vypočítává čas mezi dvěma vzorky $\triangle t$ v sekundách. Tato hodnota je povětšinou konstantní, protože vzorky jsou ekvidistantní v čase, ale vzhledem k množství operací, které musí MCU během měření synchronizovat, dochází k občasnému prodloužení nebo zkrácení měřicí periody, a proto je výpočet přesné hodnoty $\triangle t$ nezbytný. Tato informace je využívána mnoha dalšími VI, která pracují s časem nebo provádí integraci.

Mezi nejdůležitější VI volané ve smyčce programu patří *VehicleToGravityTransformation.vi* a *VehicleToHeadingTransformation.vi*. První jmenované VI provádí transformaci vektoru zrychlení měřeného analogovým senzorem do soustavy orientované vůči tíhovému zrychlení, jinými slovy, eliminuje vliv tíhového zrychlení na složky zrychlení a_x a a_y, čímž je kompenzován náklon vozidla i nepřesnosti v montáži senzoru. Druhé VI provádí také transformaci vektoru zrychlení, tentokrát ale jde o přepočet do soustavy orientované ve směru jízdy. Tento proces zajišťuje, že působení dopředného zrychlení se projevuje výhradně ve složce a_x a působení dostředivého zrychlení se promítá pouze do složky a_y, bez ohledu na to, jak přesně či spíše nepřesně je měřicí zařízení instalováno ve vozidle. Protože obě uvedené VI realizují jedny z klíčových algoritmů navigace, budou níže popsány podrobněji. V tuto chvíli je jen třeba zmínit dva logické výstupy, které z nich vycházejí. Je to signál *Stop*, který vychází z *VehicleToGravityTransformation.vi* a signál *Heading*, který je na výstupu *VehicleToHeadingTransformation.vi*. Signál *Stop* je v logické hodnotě pravda vždy, když je detekované stání vozidla, zatímco signál *Heading* nabývá hodnoty pravda v situaci, kdy vozidlo výrazně zrychluje na rovném úseku.

Jedním ze vstupních parametrů pro algoritmus transformace do směru jízdy je také aktuální rychlost vozidla. Ta je získávána integrací složky zrychlení a_x pomocí sub-VI s názvem *IntegrateVelocity.vi*. Nulování integrace[7] se provádí příslušným vstupem VI, do kterého je přiveden výše zmiňovaný signál *Stop* v logickém součtu s výstupem několika dalších podmínek. Tuto konstrukci je možné lépe popsat pomocí části kódu ukázané na obrázku 5.3. Pokud je výstup integračního VI větší než nastavená hodnota *VelocityLimit* nebo menší než nula (zatím není ošetřeno couvání) nebo pokud signál *Stop* nabývá hodnoty pravda, je integrace nulována. Z principu Dataflow [52] na kterém je programování v LabVIEW založeno, musí být výstup prvních dvou podmínek přiveden na vstup inte-

[7] Touto metodou je potlačována integrační chyba.

gračního VI přes tzv. Feedback Node. To v praxi znamená, že výsledek podmínek se na vstup dostane až v příštím iteračním kroku programové smyčky, a proto je výstup prvních dvou podmínek přiveden ještě do objektu Select, který dá na výstup hodnotu nula pokud je log. vstup pravda a výsledek integrace, pokud je log. vstup nepravda.



Obr. 5.3: Ukázka kódu LabVIEW - výpočet rychlosti integrací

Mimo výše popsané výpočty, které pracují s hodnotami získanými výhradně z analogových senzorů, obsahuje navigační program ještě implementaci strap-down algoritmu, jež byla popsána v kapitole 2.2.3.2. Hodnoty úhlových rychlostí měřené digitálním gyroskopem jsou nejprve prahovány pomocí sub-VI s názvem *TresholdBand.vi*, jehož vstupem je kromě úhlových rychlostí ω_φ, ω_θ a ω_ψ, také požadovaná prahová hodnota. Takto upravené signály dále vstupují do VI s názvem *DirectionCosines.vi*, které, jak jeho název napovídá, implementuje matematickou metodu směrových kosinů. Výstupem VI je pak transformační matice, jež vstupuje do dalšího sub-VI s názvem *Transform.vi*, kde je s její pomocí provedena rotace (maticové násobení) vektoru zrychlení.

Jediná měřená veličina, která není v navigačním programu zatím využívána, je zrychlení měřené digitálním senzorem. Důvod je ten, že při porovnání dat z digitálního akcelerometru s daty získanými analogovým akcelerometrem po jejich filtraci navrženým číslicovým filtrem, vykazují data z analogového senzoru větší přesnost a kvalitu. Tento výsledek platí prakticky pro všechna dosud provedená měření, a proto je akcelerace měřená digitálním senzorem využívána pouze pro porovnání a informativní účely.

5.3.1.2 Zpracování dat pro výstup

Tato část programu běží rovněž v hlavní smyčce a operace jsou vykonávány v každé iteraci. Jedná se vlastně o sérii podmínek, jejichž logická hodnota ovlivňuje postupnou tvorbu textového řetězce, jakéhosi itineráře popisujícího pohyb a chování navigovaného objektu. Tento textový popis je pak zobrazen po skončení celého programu a tvoří jeden z hlavních výstupů. Jednotlivé parametry, které se do itineráře zapisují je možné zapínat či vypínat pomocí přepínačů na ovládacím panelu aplikace, nastavení ale musí být provedeno před spuštěním VI. V současné době navigační systém zpracovává pro itinerář tyto informace:

- Zastavení vozidla a opětné uvedení do pohybu

- Vstup a výstup do/z definovaného prostoru
- Velikost úhlu dosaženého při průjezdu obloukem
- Poloměr projetého oblouku
- Ujetá vzdálenost mezi zastaveními

Parametry byly vybrány s ohledem na možné uplatnění popisovaného navigačního systému v konkrétní realizaci, ale v podstatě je možné vytipovat další nebo jiné parametry, které mají být výstupem systému. Mimo vlastní parametr a případně i jeho hodnotu, obsahuje každý záznam v itineráři také informaci o čase, kdy byl zapsán. Tento čas je v UTC a je aktualizován z GNSS dat.

Zastavení vozidla a opětné uvedení do pohybu je detekováno poměrně snadno, pomocí signálu *Stop*, resp. detekováním jeho náběžné a sestupné hrany pomocí sub-VI s názvem *MiscEdgeDetector.vi*. Je-li detekována náběžná hrana, do výstupního řetězce je přidán záznam obsahující časový údaj a slova „Vozidlo zastavilo“. Pokud je detekována hrana sestupná, pak je přidán text „Vozidlo se rozjelo“.

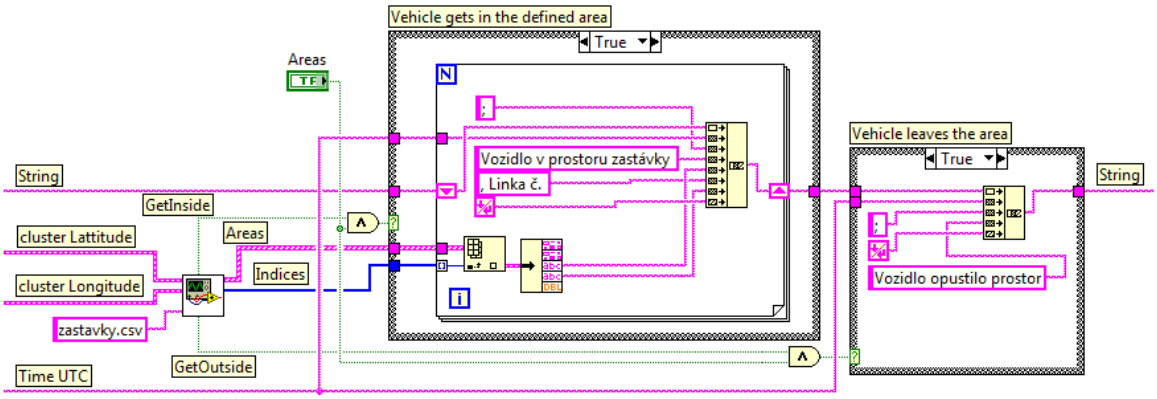
Přítomnost v definovaném prostoru je určována na základě informací o poloze (zeměpisných souřadnicích) získaných z GNSS dat. Pro snazší popis této části programu je opět použit úryvek zdrojového kódu, zobrazený na obrázku 5.4. Zeměpisné souřadnice šířka (lattitude) a délka (longitude) vstupují do sub-VI s názvem *MapPointDetection.vi*, případně *MapPointDetectionPolygon.vi*, v závislosti na verzi souboru *zastavky.csv*. Výstupem VI jsou logické hodnoty *GetInside* a *GetOutside*, které jsou hranové a které indikují zda se navigovaný objekt dostal do prostoru nebo zda jej právě opustil. Dalšími výstupy jsou pole clusterů (struktur), které obsahuje všechny prostory načtené ze souboru s jejich parametry (název zastávky, číslo linky, směr atd.) a pole indexů, které označuje prostory ve kterých se vozidlo právě nachází[8]. Logický součin signálů *GetInside* a *GetOutside* s výstupem přepínače *Areas* umožňuje aktivovat nebo deaktivovat zápis prostorů do itineráře. Zápis samotný je řešen smyčkou, jejíž počet kroků je dán délkou pole indexů, tedy počtem aktivních prostorů. Zapsán je vždy čas, text „Vozidlo v prostoru zastávky“, název prostoru a číslo linky. V případě opuštění prostoru je zapsán pouze čas a text „Vozidlo opustilo prostor“.

Pro zápis velikosti úhlu dosaženého při průjezdu obloukem je klíčové VI s názvem *PeakAngleDetection.vi*. Do VI vstupují hodnoty čas $\triangle t$, úhlová rychlost ω_ψ a prahová hodnota pro velikost úhlu, který již nemá být zanedbán. Výstupem VI je pak vlastní hodnota naintegrovaného dosaženého úhlu a také logická hodnota *Peak*, která je hranová. Zápis do textového řetězce probíhá podobně jako u detekce prostorů a text obsahuje čas, úhel ve stupních a směr vlevo nebo vpravo, určený na základě znaménka hodnoty úhlu.

Výpočet poloměru projetého oblouku a zápis výsledku do myšleného itineráře v podstatě naplňuje jeden z vytyčených cílů práce, tedy detekci charakteristických segmentů

[8]Pokud se prostory z nějakého důvodu překrývají, pak jsou vráceny indexy všech dotyčných prostorů.

trati. Klíčovou roli zde hraje algoritmus implementovaný ve VI s názvem *CalculateRadius.vi*, do něhož vstupují veličiny čas $\triangle t$, úhlová rychlost ω_ψ a vypočítaná aktuální rychlost vozidla. Dalším vstupem je prahová hodnota minimálního úhlu ve stupních a tolerance (přesnost) s jakou má být iterativním postupem porovnáván úhel natočení (viz. popis výpočtu v kapitole 3.2.1). Výstupem VI je pak vypočítaný poloměr v metrech a také logický hranový signál *IsValid*, na jehož základě probíhá zápis do itineráře, analogicky s výše uvedenými postupy. Záznam obsahuje časový údaj, text „Poloměr oblouku" a hodnotu poloměru v metrech.



Obr. 5.4: Ukázka kódu LabVIEW - detekce přítomnosti v prostoru

Pro záznam ujeté vzdálenosti mezi zastaveními platí prakticky stejný postup jako pro předchozí dva uváděné parametry a výpočet v tomto případě obstarává VI s názvem *PeakDistanceDetection.vi* jehož vstupem je naintegrovaná ujetá vzdálenost. Tuto vzdálenost počítá sub-VI s názvem *IntegrateDistance.vi* z hodnoty času $\triangle t$ a aktuální rychlosti vozidla, přičemž nulování integrátoru je navázáno na již známý signál *Stop*. Pro výpočet vzdálenosti je používána rychlost počítaná z inerciálních veličin, jak to bylo popsáno výše nebo lze alternativně použít údaj o rychlosti obsažený v GNSS datech. Možnost použití rychlosti z GNSS je podmíněna jednak platností dat, ale také nastavením příslušného přepínače na ovládacím panelu aplikace. Vzhledem k tomu, že požadavkem na celý systém je fungování bez GNSS příjmu, možnost použít rychlost ze satelitní navigace je zde implementována pouze pro účely porovnání a ladění.

5.3.1.3 Ovládací panel

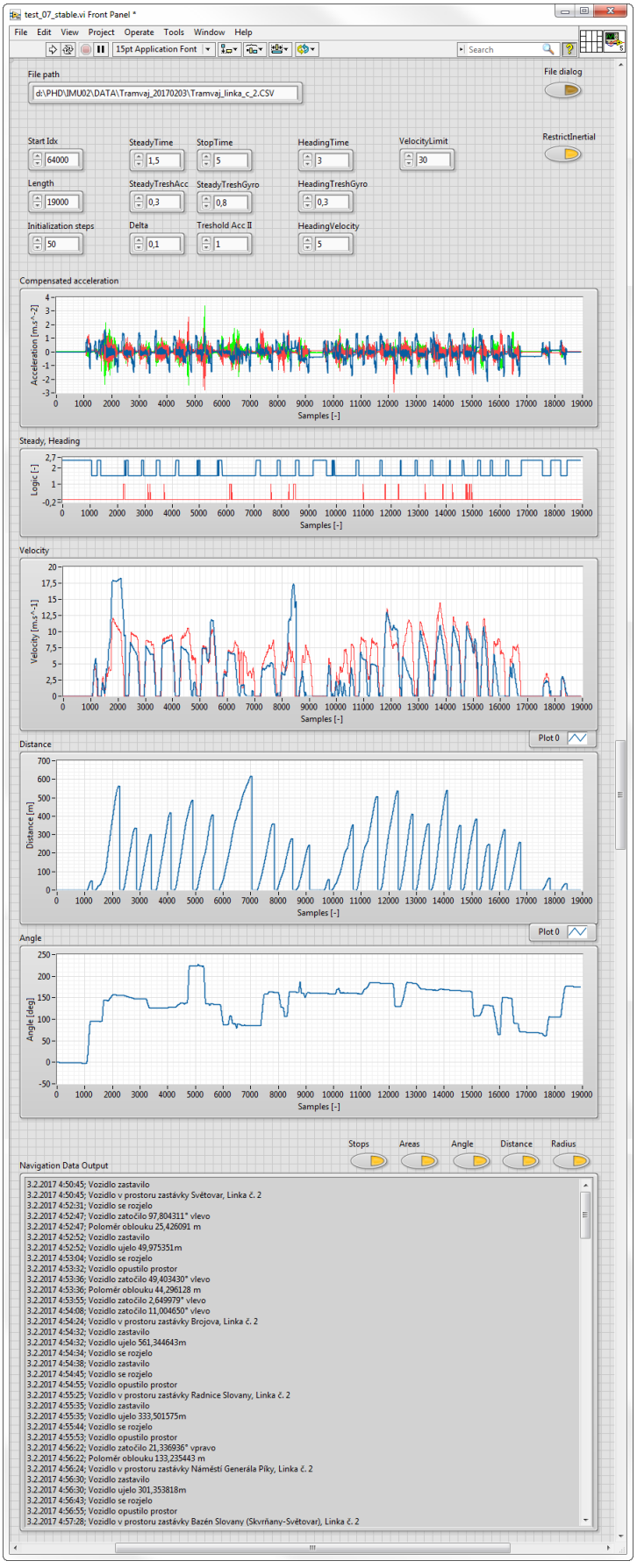
V rámci popisu funkce navigačního programu je vhodné uvés ještě několik informací o ovládacím panelu aplikace, které tvoří GUI pro práci s programem. Jak bylo uvedeno v kapitole o softwaru LabVIEW, jeho hlavní výhoda v oblasti GUI spočívá v tom, že GUI je pod názvem Front Panel, automaticky součástí každého vytvořeného VI. Přidávání tlačítek, indikátorů, grafů či prvků ActiveX je tak otázka okamžiku a navázání těchto prvků na vstupní a výstupní signály v kódu (Block Diagram) je rovněž snadné. Tyto vlastnosti umožňují během práce na programu zobrazovat libovolné hodnoty v libovolném stádiu

jejich algoritmického zpracování a jak program tak i GUI se neustále vyvíjí. Poslední ustálený stav hlavního programu je reprezentován VI s názvem *Test07Stable.vi*, respektive jeho GUI bude tedy nyní stručně popsáno.

Celý panel VI je ukázán na obrázku 5.5, včetně zachycených reálných dat z tramvajového provozu. V horní části se nachází pole *FilePath* pro zadání cesty k souboru s měřenými daty, který je programem automaticky načítán a přepínač *FileDialog*, kterým lze nastavit zobrazení standardního dialogu pro výběr souboru, namísto automatického načtení souboru ve *FilePath*. Dále následuje skupina třinácti číselných vstupů, z nichž první tři zleva slouží k nastavení počátečního indexu a délky pro omezení vstupních dat a k nastavení počtu vzorků, přes které se má provést počáteční průměrování. Dalších šest ovladačů slouží k nastavení parametrů pro algoritmus přepočtu vektoru zrychlení do soustavy orientované vůči tíhovému zrychlení. Tři parametry uprostřed panelu pak slouží pro nastavení algoritmu přepočtu vektoru zrychlení do soustavy orientované vůči směru jízdy a poslední ovladač s názvem *VelocityLimit* slouží k nastavení max. teoretické rychlosti vozidla v metrech za sekundu. Překročení této rychlosti během výpočtu indikuje integraci s příliš velkou chybou (viz. obrázek 5.3). Posledním prvkem v této části GUI je přepínač označený, který je aktivní, což znamená, že pro výpočet ujeté vzdálenosti je použita výhradně rychlost spočítaná z inerciálních veličin.

Pod ovládacími prvky v horní části GUI je dále seřazeno pod sebou několik grafů. Ve směru od shora, zachycují první graf průběh třech složek zrychlení po všech provedených transformacích s kompenzovanou tíhovou složkou, kde modrý průběh odpovídá dopředné složce a_x, červený průběh příčné složce a_y a zelený vertikální složce a_z. Druhý graf v pořadí zachycuje průběhy dvou logických signálů, kde horní (modrý) průběh zachycuje signál *Stop* a spodní (červený) pak signál *Heading*. Třetí graf zobrazuje rovněž dva průběhy, a sice průběhy dopředných rychlostí. Modrý průběh vyjadřuje rychlost spočítanou navigačním programem pouze z inerciálních veličin a červený průběh zobrazuje rychlost získanou z GNSS přijímače. Porovnáním obou průběhů pak lze částečně validovat nastavení navigačních výpočtů. Další, předposlední zobrazovaný graf neukazuje přímo ujetou vzdálenost, ale je možné z grafu určit délky jednotlivých úseků mezi zastaveními vozidla. Tato úseková vzdálenosti je dána hodnotou lokálních maxim zobrazeného průběhu, přičemž místa s nulovou hodnotou odpovídají okamžikům, kdy vozidlo stálo. Poslední graf na panelu je pouze informativní a ukazuje průběh úhlu natočení vozidla během celé jízdy, který je počítaný integrací úhlové rychlosti ω_ψ bez nulování.

Pod zobrazenými grafy se v pravé části GUI nachází celkem pět přepínačů označených *Stops*, *Areas*, *Angle*, *Distance* a *Radius*. Pokud je některý přepínač aktivní (na obrázku jsou aktivní všechny), je odpovídající parametr jízdy zapisován do itineráře. Právě tento itinerář, resp. textový výstup navigačního programu je zobrazen v poli pod přepínači, které je označeno *NavigationDataOutput*.



Obr. 5.5: Ukázka GUI hlavního navigačního programu

5.3.2 Sub-VI realizující klíčové algoritmy

Během popisu navigačního programu byla zatím podrobně popsána jeho funkcionalita a struktura hlavního VI, včetně GUI a výstupů. V popisu byla také zmíněna některá důležitá sub-VI, která jsou hlavním programem volána. Vzhledem k tomu, že tyto podprogramy realizují důležité matematické operace a algoritmy, budou některé z nich v následujícím textu detailněji popsány.

5.3.2.1 Načtení souboru s naměřenými daty a rozdělení do polí vzorků

VI s názvem *ReadDataFile.vi*, zajišťující načtení CSV souboru s naměřenými daty, využívá vestavěná sub-VI pro práci se soubory a pro zpracování řetězců a je tedy poměrně jednoduché. Jeho funkce spočívá v přečtení prvního řádku ze souboru, ten se ale nezpracovává a následného čtení všech zbývajících řádků. Každý řádek je pak zpracován ve smyčce, pomocí sub-VI pro hledání tokenů dle zadaného delimiteru, v tomto případě středníku. Pole řetězců (tokenů), jehož délka odpovídá počtu sloupců v CSV souboru, je následně zpracováno pro každý z jeho prvků odlišně. První prvek je pouze převeden z textu na číslo, neboť jde o časový údaj, druhý token je komplexnější řetězec a obsahuje GNSS data. GNSS záznam je zpracován pomocí VI s názvem *GpsGPRMC.vi* a výstupem je cluster (struktura) obsahující informaci o platnosti dat (logická hodnota), UTC čas (speciální časová struktura LabVIEW), zeměpisnou šířku a délku (struktury obsahující hodnotu ve stupních a znak určující zeměpisný směr), rychlost vůči zemskému povrchu (hodnota v uzlech), zeměpisný kurs (hodnota ve stupních) a některé další údaje.

Ostatní načtené tokeny již obsahují pouze číselné hodnoty. Ty jsou tedy převedeny na čísla a protože jsou bez rozměru, je také proveden přepočet na fyzikální veličiny, případně ještě posunutí nuly[9]. Pole načtených a přepočítaných hodnot jsou dále, podle typu obsažených dat, sdružena do clusterů *AnalogSensors* a *DigitalSensors*. Toto opatření je provedeno zejména pro přehlednost. Výstupem VI jsou tedy data uspořádaná dle následujících bodů:

- Pole hodnot typu double s hodnotami času
- Pole struktur s GNSS daty
- Cluster *AnalogSensors* obsahující pět polí hodnot typu double pro složky akcelerace a_x, a_y a a_z měřené analogovým akcelerometrem a referenční a výstupní hodnotu ω_ψ analogového gyroskopu
- Cluster *DigitalSensors* obsahující šest polí hodnot typu double pro složky akcelerace a_x, a_y a a_z měřené digitálním akcelerometrem a úhlové rychlosti ω_φ, ω_θ a ω_ψ měřené digitálním gyroskopem
- Pole hodnot typu double s hodnotami napětí baterie

[9]Hodnoty v měřicím souboru jsou v celočíselném formátu s pevnou řádovou čárkou.

- Pole hodnot typu double s hodnotami teploty

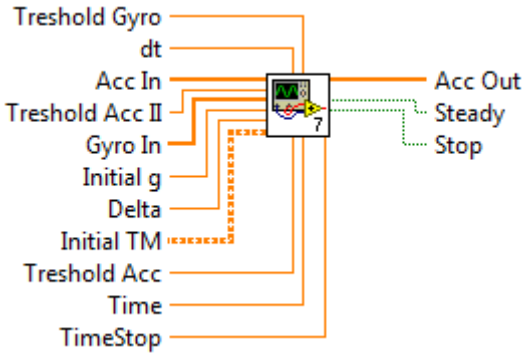
Pro úplnost zbývá uvést, že hodnoty času jsou v milisekundách, akcelerace je v metrech za sekundu na druhou, úhlová rychlost je ve stupních za sekundu, napětí baterie ve voltech a teplota ve stupních Celsia.

5.3.2.2 Výpočet úhlů klonění a klopení a transformační matice

Výpočet úhlů klonění a klopení a transformační matice je realizován pomocí VI s názvem *AccToPitchAndRoll.vi*. Vstupem jsou zde pouze tři složky zrychlení, ze kterých jsou přímou implementací rovnic (3.1) a (3.2) (viz. kapitola 3.1.1) spočítány úhly klonění (roll) a klopení (pitch) v radiánech. Následně jsou spočítány sinové a kosinové složky a je poskládána transformační matice tak, jak je definována v rovnici (3.3) v kapitole 3.1.1. Výstupem VI je pak vypočítaná transformační matice i oba úhly, přepočítané na stupně.

5.3.2.3 Přepočet vektoru zrychlení do soustavy orientované ve směru tíhového zrychlení

Přepočet vektoru zrychlení je řešen ve VI s názvem *VehicleToGravityTransformation.vi*, jehož vstupy a výstupy jsou ukázány na obrázku 5.6. VI implementuje stavový automat, neboť si musí pamatovat stav mezi jednotlivými voláními. Toto se v LabVIEW řeší tak, že celý kód VI je vložen do smyčky typu while, jejíž terminál pro zastavení je trvale spojen s konstantou v logické hodnotě pravda. Smyčka se tedy při každém volání vykoná pouze jedenkrát, ale na jejím okraji je možné pomocí terminálu typu Shift Register uchovávat předchozí stavy libovolných proměnných. Inicializace stavů a proměnných je pak zajištěna podmínkovou strukturou Case Structure na jejíž logický vstup je připojen výstup prvku First Call. Nastavení jednotlivých proměnných pro první volaní VI je zajištěno příslušným kódem uvnitř této podmínky.



Obr. 5.6: Vstupy a výstupy *VehicleToGravityTransformation.vi*

Na začátku je algoritmus ve stavu *Idle*, všechny pomocné proměnné jsou vynulované a výstupní signály *Stop* a *Steady* jsou v logické hodnotě pravda[10]. Vstupní vektor zrychlení

[10]Předpokládá se, že při prvním volání algoritmu vozidlo stojí.

Acc In je transformován inicializační hodnotou transformační matice *Initial TM* a je kompenzována tíhová složka, v této fázi pomocí vstupní hodnoty *Initial g*. Upravené zrychlení je jednak předáno na výstup jako pole (vektor) *Acc Out*, ale je také uloženo do stavové proměnné tak, aby tato hodnota byla k dispozici v příštím volání VI.

Nyní následuje řešení podmínek, které řídí stavový automat. Nejprve je porovnána absolutní hodnota každé složky transformovaného zrychlení s prahovou hodnotou *Treshold Acc*, zda je menší nebo rovna. Zároveň je porovnána absolutní hodnota vstupu *Gyro In* s hodnotou *Treshold Gyro*, zda je také menší nebo rovna. Výstup obou podmínek je spojen v logickém součinu a dále v logickém součtu s hodnotou výstupního signálu *Stop* (je to zároveň stavová proměnná). Význam výstupu podmínek je v tomto bodě následující: pravda - vozidlo zastavilo nebo již stálo, nepravda - hodnoty akcelerace a úhlové rychlosti odpovídají stavu, kdy je vozidlo v pohybu. Pokud je tedy logická hodnota pravda, jsou vynulovány čítač vzorků, čítač času a akumulátor hodnot zrychlení a stav automatu je změněn na *Detect*.

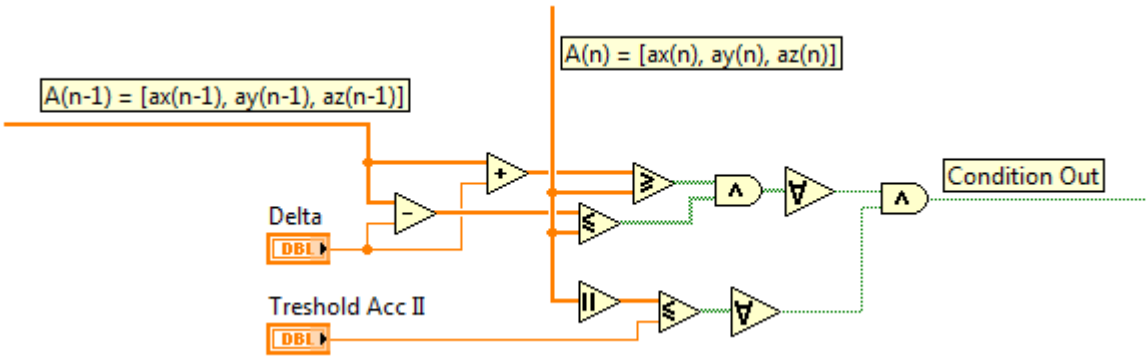
Ve stavu *Detect* je transformace vektoru zrychlení i série výše uvedených podmínek řešena stejně jako ve stavu *Idle*, jen navíc, pokud je podmínka stání stále splněna, jsou hodnoty vstupního zrychlení (tedy netransformované a bez kompenzace tíhového zrychlení) přičítány do akumulátoru, čítač vzorků je inkrementován a k čítači času je přičítána vstupní hodnota *dt*. Pokud se vozidlo v tuto chvíli opět rozjede, je podmínka stání porušena a algoritmus přechází zpět do stavu *Idle*. Zůstane-li vozidlo v klidu až do chvíle, kdy hodnota čítače času je větší nebo rovna vstupní hodnotě *Time*, přechází automat do posledního svého stavu s názvem *Steady* a zároveň je nastaven výstupní signál *Steady* do logické hodnoty pravda.

Pokud je podmínka stání stále splněna, probíhá načítání hodnot zrychlení a inkrementace čítače vzorků i ve stavu *Steady*. Je-li v této fázi vozidlo opět uvedeno do pohybu a podmínka je tím porušena, je zkontrolován stav čítače vzorků, zda je jeho hodnota větší než nula, a pokud ano, jsou vypočítány průměrné hodnoty všech tří složek zrychlení. Výpočet je proveden jednoduše, dělením hodnot nasčítaných v akumulátoru, hodnotou čítače vzorků. Výsledek je následně předán do VI s názvem *AccToPitchAndRoll.vi* a to vrátí novou transformační matici, která je zapamatována a používána dokud se celý cyklus automatu neopakuje. V tomto místě je také uložena nová hodnota tíhového zrychlení, která je následně používána pro kompenzaci. Poslední operací je pak opětovné uvedení signálu *Steady* do hodnoty nepravda a návrat do výchozího stavu *Idle*.

Důležitým výstupem popisovaného VI je již několikrát zmiňovaný signál *Stop*. Ten, na rozdíl od signálu *Steady*, který signalizuje jakékoli detekované zastavení, vyjadřuje zastavení v zastávce, resp. takové zastavení vozidla, jehož trvání překračuje nastavený čas *TimeStop*. Zpracování signálu *Stop* je řešeno paralelně s výše popsaným stavovým automatem, za pomoci několika dalších pamatovaných proměnných a série podmínek.

Již bylo uvedeno, že transformovaný vektor zrychlení je ukládán tak, že v každém volání VI je k dispozici jeho předchozí hodnota. Na obrázku 5.7 je ukázáno několik podmí-

nek, které jsou vyhodnocovány kolem předchozí a současné hodnoty zrychlení. Výstupem podmínek je logická hodnota jež určuje, zda jsou nové hodnoty složek zrychlení v delta okolí jejich předcházejících hodnot (definováno vstupem *Delta*) a současně, jejich absolutní hodnoty jsou menší než vstupní hodnota *Treshold Acc II*. Je-li podmínka splněna, načítá algoritmus při každém volání VI do pomocné proměnné čas *dt* a výsledek porovnává s hodnotou *TimeStop*. Po překročení hodnoty je pak signál *Stop* uveden do stavu pravda a je vynulován další pomocný čítač. Není-li podmínka z obrázku 5.7 splněna, drží signál *Stop* stále svoji předchozí hodnotu a pomocný čítač je inkrementován při každém volání programu. Jeho hodnota je opět porovnávána se vstupem *TimeStop* a ve chvíli, kdy hodnota čítače překročí tento čas, signál *Stop* je uveden do stavu nepravda. Celý popsaný mechanismus kolem signálu *Stop* zaručuje, že detekování zastávky je spolehlivé a neprojeví se zde náhodné výkyvy v měřených datech, či nekonzistence v pohybu vozidla.



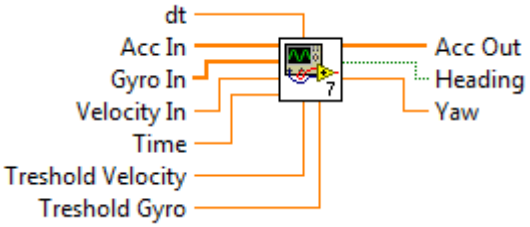
Obr. 5.7: Ukázka kódu LabVIEW - podmínky pro detekování zastávky

5.3.2.4 Přepočet vektoru zrychlení do soustavy orientované ve směru jízdy vozidla

Algoritmus přepočtu vektoru zrychlení do směru jízdy, realizovaný v sub-VI s názvem *VehicleToHeadingTransformation.vi*, je řešen velmi podobně, jako předchozí popsané VI. Seznam vstupních a výstupních signálů je patrný z obrázku 5.8, přičemž kromě vlastního transformovaného vektoru zrychlení *Acc Out*, je důležitým výstupem také logický signál *Heading*, který indikuje přímou jízdu vpřed s výraznou hodnotou zrychlení.

Stavový automat je zde realizován známým způsobem a vnitřní stavové proměnné jsou inicializovány během prvního volání VI. Rozdílem proti předchozímu VI je inicializace výchozí transformační matice, která zde není brána ze vstupu, ale je nastavena konstantou o hodnotě jedna, přesněji jednotkové matice, což v praxi znamená, že počáteční rotace je nulová. Maticové násobení vstupního vektoru zrychlení s touto maticí pak probíhá v každém volání VI a transformovaná hodnota je k dispozici na výstupu. Z výchozího stavu *Idle* přechází algoritmus do stavu *Detect* na základě splnění podmínky, kdy absolutní hodnoty vstupních úhlových rychlostí *Gyro In* musí být menší nebo rovny vstupu *Treshold Gyro*

a současně absolutní hodnota rychlosti *Velocity In* je větší nebo rovno vstupu *Treshold Velocity*. Při přechodu do stavu *Detect* je také vynulován akumulátor hodnot zrychlení, čítač vzorků a čítač času.



Obr. 5.8: Vstupy a výstupy *VehicleToHeadingTransformation.vi*

Ve stavu *Detect* jsou sčítány hodnoty složek zrychlení jako netransformované vstupní hodnoty, ale je předpokládáno jejich předcházející zpracování algoritmem pro transformaci do souřadného systému orientovaného ve směru tíhového zrychlení. Dále je inkrementován čítač vzorků a sčítány vstupní časy *dt*. To vše je samozřejmě podmíněno trvajícím splněním podmínky popsané výše a pokud tato splněna není, algoritmus přechází zpět do stavu *Idle*. Je-li podmínka splněna až do chvíle, kdy je nasčítaný čas větší nebo roven vstupní hodnotě *Time*, přechází automat do stavu *Heading* a výstupní signál téhož názvu je nastaven do logické hodnoty pravda.

Až do porušení podmínky pro rychlost, probíhá sčítání hodnot a inkrementace čítače i ve stavu *Heading*. Když vozidlo zpomalí, nebo začne zatáčet, podmínka již není splněna a proběhne výpočet nové transformační matice, opět za pomoci vhodně použitého VI s názvem *AccToPitchAndRoll.vi*. Protože výsledek rotace v prostoru je závislý na pořadí provedení jednotlivých elementárních rotací, je v tomto případě vhodným použitím VI myšleno správné pořadí složek ve vstupním vektoru zrychlení. K výpočtu jsou použity průměrované složky zrychlení a provedení výpočtu je též podmíněno nenulovou hodnotou čítače. Produktem výpočtu je též aktuální hodnota úhlu vyjadřujícího odchylku os akcelerometru od os ve kterých působí složky zrychlení na vozidlo, v rovině xy. Hodnota úhlu ve stupních je uložena do stavové proměnné, která je připojena na výstup *Yaw*. Po provedení této části programu se automat vrací zpět do výchozího stavu *Idle* a signál *Heading* nabývá hodnoty nepravda.

5.3.2.5 Implementace metody směrových kosinů pro strap-down algoritmus

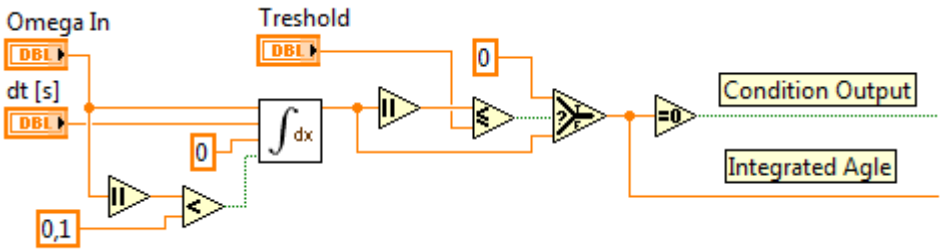
Problematika směrových kosinů je řešena ve VI s názvem *DirectionCosines.vi* a jde v podstatě o přímou implementaci rovnic (2.8) až (2.17) uvedených v kapitole 2.2.3.2. Vstupem VI jsou tři hodnoty úhlových rychlostí, které jsou nejprve testovány na nulovou hodnotu a další část algoritmu je provedena jen tehdy, je-li aspoň jedna složka nenulová. Dále jsou hodnoty převedeny na jednotky radiány za sekundu a přenásobeny vstupním časovým intervalem *dt*. Ze získaných součinů a případně také jejich negovaných hodnot je následně poskládána matice $\boldsymbol{B}$ a vypočítána hodnota σ. Nakonec je z matice $\boldsymbol{B}$, jejího kvadrátu,

jednotkové matice a hodnoty σ, poskládána rovnice pro transformační matici $\boldsymbol{C}(t + \triangle t)$, se kterou je maticově násobena vstupní matice *Matrix In* a výsledek je předán na výstup jako *Matrix Out*.

5.3.2.6 Detekování a výpočet dosaženého úhlu během jednoho souvislého otáčení

Výpočet dosaženého úhlu při otáčení, respektive vlastní hodnota úhlu je jedním z výstupů navigačního algoritmu, v rámci detekce charakteristických segmentů trati. Algoritmus je realizován v sub-VI s názvem *PeakAngleDetection.vi* a jeho implementace opět vede na stavový automat.

Jednou z hodnot, která podmiňuje změnu stavu automatu je výstup podmínky uvedené na obrázku 5.9. Základem úryvku kódu je integrátor, sub-VI s názvem *MiscIntegrate.vi*, který integruje vstupní vzorky úhlové rychlosti *Omega In* s časovým krokem daným vstupem *dt*. Nulování integrátoru nastává ve chvíli, kdy je absolutní hodnota úhlové rychlosti menší než nastavená prahová hodnota. Výstup integrátoru, úhel, je dále prahován na základě vstupu *Treshold* tak, že když je jeho absolutní hodnota menší nebo rovna té prahové, je hodnota úhlu nahrazena nulou. Tento mezivýsledek je na obrázku označen *Integrated Angle* a je dále zpracováván ve VI. Výstup podmínky, logická hodnota *Condition Output*, je nakonec získána porovnáním vypočítaného úhlu s nulou.



Obr. 5.9: Ukázka kódu LabVIEW - podmínka vyhodnocení integrace úhlu

Je-li výsledek uvedené podmínky pravda, zůstává algoritmus ve stavu *Idle*. V opačném případě, když je úhel nenulový, se stav mění buď na *Up* pro kladné hodnoty úhlu nebo *Down* pro záporné. Zároveň je absolutní hodnota úhlu zapamatována pro další volání VI. Ve stavu *Idle* je výstupní signál indikující dosažení maxima úhlu (*Peak*) držen ve stavu nepravda.

Ve stavech *Up* i *Down* je opět kontrolována podmínka nenulovosti úhlu a pokud je splněna, je absolutní hodnota úhlu porovnávána s předchozí hodnotou a větší z nich je pamatována. Signál *Peak* je stále aktivně držen ve stavu nepravda. Nastane-li nulování integrátoru a následně, také úhel dosáhne nulové hodnoty, přechází algoritmus zpět do výchozího stavu *Idle*, přičemž do výstupní hodnoty *Angle Out* je zapsána poslední pamatovaná hodnota úhlu, resp. tato hodnota negovaná v případě aktivního stavu *Down*.

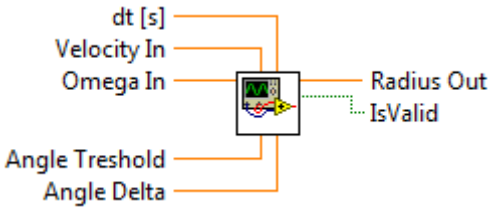
Signál *Peak* je v tomto okamžiku nastaven na hodnotu pravda. Důležitou vlastností tohoto signálu je jeho hranový charakter, což je využíváno v nadřazeném VI.

5.3.2.7 Výpočet souvisle ujeté vzdálenosti

Určování souvisle ujeté vzdálenosti funguje na stejném principu jako právě popsaný výpočet úhlu a algoritmus je naprogramován ve VI s názvem *PeakDistanceDetection.vi*. Vstupem tohoto VI je přímo integrovaná vzdálenost, takže realizace podmínky je ještě jednodušší než v případě výpočtu úhlu. Vzdálenost je prahována vstupem *Treshold* a výstup, dosažená vzdálenost, je nazván *Distance Out*. Signál *Peak* má stejný význam jako v předchozím VI a je též hranový.

5.3.2.8 Výpočet poloměru projetého oblouku

Algoritmizace výpočtu poloměru oblouku vedla asi na nejsložitější sub-VI naprogramované v rámci celého projektu. VI se jmenuje *CalculateRadius.vi* a implementuje v podstatě dva paralelní stavové automaty, které mají pouze dva stavy. První automat řeší vlastní výpočet poloměru projetého oblouku, zatímco druhý připravuje výstupní data a hranový signál *IsValid* indikující platnost výstupních dat. Tento signál je důležitý pro fungování nadřazeného VI, jak bylo popsáno výše. Pro představu o všech vstupech a výstupech se kterými popisované VI pracuje, je na obrázku 5.10 uvedena jeho schematická značka.

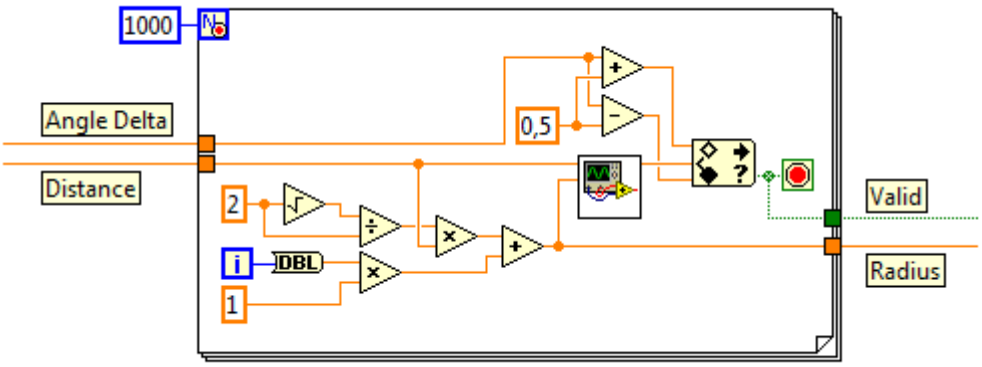


Obr. 5.10: Vstupy a výstupy *CalculateRadius.vi*

Během prvního volání VI jsou nastaveny výchozí hodnoty stavových proměnných a oba automaty jsou uvedeny do stavu *Idle*. Nezávisle na stavu, je v každém volání programu prováděna integrace vstupní úhlové rychlosti (signál *Omega In*) v čase daným vstupem *dt*, za účelem získání hodnoty úhlu o který se vozidlo otočilo. Nulování integrátoru (sub-VI s názvem *MiscIntegrate.vi*) je provedeno vždy, když absolutní hodnota úhlové rychlosti klesne pod nastavenou mez. Ve stavu *Idle* je prvním automatem porovnávána naintegrovaná hodnota úhlu, resp. její absolutní hodnota, se vstupním signálem *Angle Treshold*, který definuje minimální velikost úhlu použitelného pro výpočet. Je-li absolutní hodnota úhlu větší nebo rovna nastavenému prahu, přechází automat do stavu *Rise* a aktuální hodnota úhlu je uložena do stavové proměnné *Initial Angle*.

Ve stavu *Rise* začně algoritmus integrovat také vstupní parametr *Velocity In*, který představuje rychlost vozidla, takže výsledkem integrace je ujetá vzdálenost. Nulování tohoto integrátoru je napojeno na proměnnou *Reset Int*, resp. její minulou hodnotu uloženou

na hraně smyčky VI. Dále je řešeno několik podmínek, které kontrolují zda se neustále integrovaná hodnota úhlu vzdálila od uložené *Initial Angle* o hodnotu danou vstupem *Angle Delta* v kladném či záporném směru. Jinými slovy, zda došlo k přírůstku nebo úbytku úhlu o definovanou hodnotu. Pokud se úhel dostatečně změnil a podmínka je tedy splněna, spustí se cyklus s nastaveným počtem iterací, ve kterém je spočítána hodnota poloměru dílčího oblouku za pomoci výpočtů uvedených v kapitole 3.2.1. Pro snadnější popis je na obrázku 5.11 ukázán úryvek kódu, zachycující právě výpočtovou smyčku.



Obr. 5.11: Ukázka kódu LabVIEW - iterační výpočet poloměru oblouku

Jak bylo uvedeno v kapitole 3.2.1, hledaný poloměr je z intervalu $\left\langle \frac{\sqrt{2}}{2}d; \infty \right)$. Vstupní hodnota *Distance* (vzdálenost získaná integrací rychlosti) je tedy přenásobena $\frac{\sqrt{2}}{2}$ a k této hodnotě je přičítáno pořadové číslo iterace i. Výsledek, hledaný poloměr, pak společně se vzdáleností *Distance* vstupují do sub-VI s názvem *CalculateBeta.vi*, ve kterém je implementována rovnice 3.12 z kapitoly 3.2.1. Výstupem VI je úhel ψ, který je testován, zda leží v definovaném intervalu kolem hodnoty *Angle Delta*. Tímto způsobem je postupně procházen celý interval a dojde-li ke shodě, znamená to, že hodnota poloměru je nalezena. Výstupem smyčky je jak samotná hodnota poloměru *Radius*, tak i logická hodnota *Valid*, která indikuje úspěšný výpočet a platnost hodnoty poloměru. Poslední operací v této části algoritmu je uložení logické hodnoty pravda do stavové proměnné *Reset Int* tak, aby došlo k nulování integrátoru rychlosti během příštího volání VI.

Poslední podmínka, která je vyhodnocována ve stavu *Rise*, porovnává absolutní hodnotu integrovaného úhlu s parametrem *Angle Treshold* a pokud je hodnota úhlu menší než prahová, algoritmus přechází zpět do stavu *Idle*. Tento mechanismus vlastně reaguje na nulování integrátoru úhlové rychlosti, které bylo zapříčiněno jejím poklesem k nule, patrně způsobeným ukončením zatáčení vozidla.

Z popisu prvního stavového automatu plyne, že hodnota poloměru oblouku je určována pro každé dílčí otočení o definovaný úhel *Angle Delta*. Druhá část VI musí tedy ze signálů *Radius* a *Valid* a dalších informací spočítat výslednou hodnotu poloměru, která je pak vrácena jako výstup celého VI. Druhý stavový automat opět začíná ve stavu *Idle*, který přetrvává až do příchodu signálu *Valid*, indikujícího nalezení hodnoty poloměru. Po

tuto dobu je výstupní signál *IsValid* držen v hodnotě nepravda. Je-li nalezení poloměru indikováno, přechází automat do stavu *Count* a jsou vynulovány dvě pomocné proměnné.

Přetrvává-li logická hodnota pravda na signálu *Valid* i ve stavu *Count*, je při každém volání VI do první pomocné proměnné přičítána hodnota poloměru a druhá pomocná proměnná je pouze inkrementována. Tato situace nastává, pokud je celkový úhel otočení větší než hodnota *Angle Delta* a je tedy během zatáčky vypočítáno více poloměrů dílčích oblouků. Když zatáčení skončí a signál *Valid* se změní na hodnotu nepravda, jsou hodnoty obou pomocných proměnných testovány na nenulovost a pokud jsou obě větší než nula, je z nich vypočítána průměrná hodnota poloměru. Tato hodnota je následně předána na výstup jako signál *Radius Out* a výstupní signál *IsValid* je nastaven na hodnotu pravda. Automat pak přechází zpět do stavu *Idle* a v dalším volání VI je signál *IsValid* nastaven zpět na hodnotu nepravda. Tím je zajištěno hranové chování tohoto signálu, což je využito v nadřazeném VI.

5.3.2.9 Detekce zda se navigovaný objekt nachází v definovaném prostoru

Detekce prostorů v nichž se navigovaný objekt aktuálně nachází je pro popisovaný navigační systém, resp. pro jeho případnou aplikaci, asi stejně zásadní funkce jako rozpoznávání segmentů trati. Stejně jako v případě segmentů, i zde je základem určitá databáze definovaných prostorů, jež jsou algoritmem periodicky testovány na přítomnost navigovaného vozidla. Jak již bylo zmíněno výše, byla vytvořena dvě VI, která řeší detekci prostorů. První VI má název *MapPointDetection.vi* a pracuje s prostory definovanými na mapě kruhem, zadaným zeměpisnými souřadnicemi jeho středu a poloměrem v metrech. Toto VI bylo vytvořeno jako první, v rámci urychlení testování, neboť tvorba databáze prostorů je relativně snadná. Druhé VI, které řeší tuto problematiku se nazývá *MapPointDetectionPolygon.vi* a jak jeho název napovídá, pracuje s databází prostorů definovaných polygonem, konkrétně čtyřúhelníkem. Toto řešení je vzhledem k charakteru testovaných prostorů (tramvajová zastávka, křižovatka apod.) výhodnější a přináší přesnější výsledky. Na druhou stranu je však tvorba databáze prostorů náročnější a vyžaduje zadat čtyři zeměpisné souřadnice pro jeden prostor. Souřadnice bodů pro polygony, resp. středy kruhů, byly z kapacitních důvodů řešeny pouze vyčítáním zeměpisných souřadnic z webových map OSM. Toto řešení pro současnou, vývojovou fázi navigačního systému postačuje, ale pro praktické nasazení by nepřesnosti v odečítání z fotografických podkladů OSM mapy, mohly znehodnocovat celou navigaci. Databáze bodů by tedy měla být profesionálně zaměřena geodeticky v systému WGS84.

Jelikož je funkce obou výše uvedených VI velmi podobná, bude jejich popis proveden společně. Po startu VI je načten obsah souboru zadaného cestou ve vstupu *FilePath*. Soubor je ve formátu CSV a v případě kruhových prostorů obsahuje zeměpisnou šířku středu kruhu ve stupních a příslušný směr (N nebo S), zeměpisnou délku středu ve stupních a příslušný směr (E nebo W), poloměr v metrech, název prostoru a název linky. V případě polygonů jsou načteny souřadnice bodů A, B, C a D, název prostoru, název linky a URL

odkaz na server OSM. Nutno dodat, že s ohledem na kompatibilitu je používán stejný CSV soubor pro oba typy VI (kruhové prostory, polygony), pouze jsou načítány odlišné sloupce. Data jsou načtena pouze při prvním volání VI a následně uložena do stavové proměnné, takže jsou k dispozici během všech následujících volání. Na výstupu VI jsou pak k dispozici jako pole clusterů s názvem *Points*.

Vstupem VI jsou dva clustery s názvy *Latitude In* a *Longitude In*, které obsahují zeměpisné souřadnice na kterých se navigovaný objekt právě nachází. Následující část VI běží ve smyčce, jejíž počet kroků je dán počtem definovaných objektů (délkou pole) a výkonný kód smyčky se liší v závislosti na typu prostorů. V případě prostorů definovaných kruhem, je nejprve zeměpisná šířka a délka středu testovaného prostoru přepočítána ze stupňů na délkové metry (viz. rovnice (2.20) a (2.21) v kapitole 2.4.2) a získanými hodnotami je vydělen poloměr kruhu. Dvě čísla vzniklá dělením pak vlastně představují malou a velkou poloosu elipsy, která ve stupních reprezentuje stejný prostor jako původní kruh zadaný poloměrem v metrech. Z rozměrů je sestavena rovnice elipsy do níž je následně dosazen rozdíl vstupní zeměpisné šířky a zeměpisné šířky středu kruhu a rozdíl vstupní zeměpisné délky a zeměpisné délky středu kruhu. Pokud je rovnice, resp. nerovnice elipsy splněna, je index testovaného prostoru přidán do pole indexů a je testován další prostor. Po skončení celé smyčky je pak pole indexů předáno na výstup *Indices* tak, aby jej mohlo využít nadřazené VI, jak bylo popsáno dříve.

Řešení smyčky pro VI pracující s polygony je v zásadě jednodušší, protože se se souřadnicemi celou dobu pracuje ve stupních a výše popsaný přepočet na délku v metrech tak není nutný. Pro vlastní testování je pak použito sub-VI z knihoven LabVIEW s názvem *Point in Polygon*, které implementuje geometrickou metodu testu. Vstupem tohoto testu jsou tedy čtyři souřadnice definující prostor a souřadnice testovaného bodu.

Mimo pole s prostory a pole indexů, poskytují obě popsaná VI také výstupní signál *Inside*, který indikuje přítomnost v jednom nebo více prostorů a dva užitečnější, hranové signály *GetInside* a *GetOutside*. Tyto hranové signály jsou tvořeny tak, že v každém volání VI je porovnávána délka pole indexů s jeho délkou během volání minulého. Při nárůstu délky je tak aktivován signál *GetInside* a při zkrácení délky pole pak signál *GetOutside*.

6

Měření a dosažené výsledky

V této kapitole bude popsána jedna z nedůležitějších částí celé práce, kterou představují výsledky dosažené navrženým navigačním systémem. Výsledky byly vytvořeny na základě série výstupů systému získané po zpracování naměřených dat. Výstupy byly zpracovány, porovnávány se skutečnými hodnotami a statisticky vyhodnoceny. Prezentace tohoto vyhodnocení v podobě grafů a procentuálních čísel pak poskytuje přehlednou a ucelenou informaci o vlastnostech a především přesnosti navržených algoritmů a o aplikovatelnosti celého systému vůbec.

Mimo zpracovaná výstupní data a verifikaci navigačního systému, obsahuje tato kapitola také informace o prováděných měřeních, ať již jde o testy v samých počátcích s prvním prototypem nebo organizovaná systematická měření v tramvajích s finálním zařízením.

6.1 První pokusy a ověřování

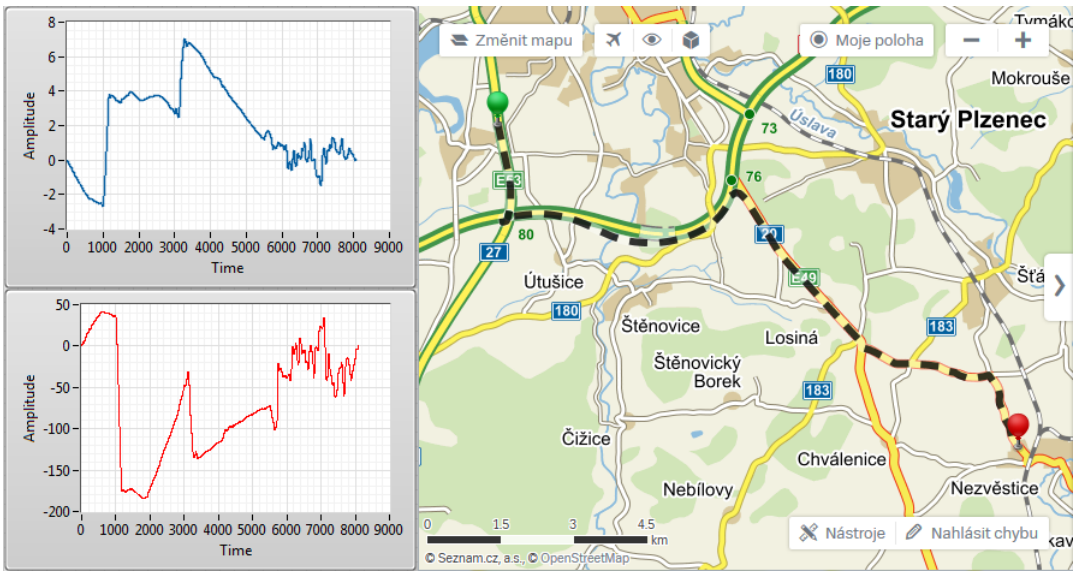
Tak jak byl výše popsán vývoj hardwaru od prvního prototypu ke druhému, vyvíjelo se i prováděné měření. Nejprve šlo o testování reálnosti řešení a získávání prvních dat, později šlo o získávání poznatků o tom, jak se které jízdní prvky a charakteristiky trasy, promítají do měřených dat a na výstupy aplikovaných výpočtů a algoritmů.

6.1.1 Měření v automobilu

Měření v automobilu patří k naprostým počátkům práce před několika lety. Sběr dat byl prováděn s prvním prototypem zařízení a bez použití GNSS přijímače. Během této fáze šlo o testování samotných senzorů a také číslicového filtru popsaného v kapitole 2.2.2.2 a jízda automobilem představovala nejsnazší způsob jak získat reálná data.

Vyhodnocení dat bylo prováděno v prostředí LabVIEW a příklad výstupu tohoto zpracování je uveden na obrázku 6.1. Obrázek je složen ze snímku dvou grafů z čelního panelu VI a snímku z webové mapy, kde je znázorněna odpovídající trasa. Horní modrý průběh pak odpovídá rychlosti a spodní červený úhlu otočení. Obě hodnoty jsou zachyceny

v závislosti na čase (čísla vzorků na vodorovné ose) a nejsou v měřítku. Přesto je ale vidět určitá korelace průběhů s trasou, zejména v místech extrémů.



Obr. 6.1: Vypočítaná rychlost a úhel natočení v porovnání s trasou během měření v automobilu

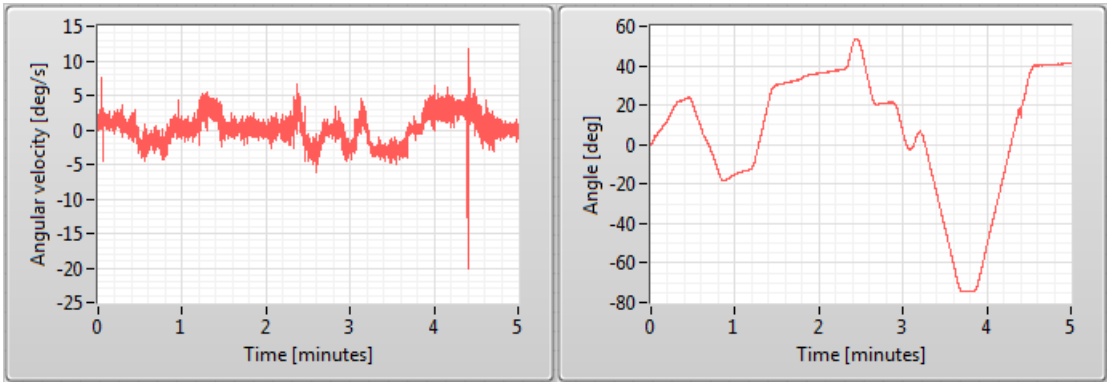
Výsledky dosažené během těchto prvních testů naznačovaly, že další vývoj v této oblasti má smysl a na jejich základě začaly vznikat první algoritmy zpracování dat. Tehdy ještě ale nepracovaly vzorek po vzorku (on-line systém), ale byly prováděny nad kompletním souborem dat se znalostí jeho délky.

6.1.2 Měření ve vlaku

Od určité doby, začal vývoj navigačního algoritmu definitivně směřovat k aplikaci pro kolejová vozidla, což bylo dáno především zájmem ze strany komerčních partnerů. Původní plány v rámci železniční dopravy nakonec však nebyly realizovány, ale některé myšlenky a princip detekce charakteristických segmentů byly přeneseny do další práce.

Měření na železnici probíhalo poměrně krátce a pouze na trati Praha - Plzeň a zpět. Během měření byla využívána první verze měřicího zařízení společně s notebookem a USB přijímačem GNSS a počítač s připojenými zařízeními byl umístěn na sedadle v prostoru pro cestující. Naměřená a uložená data pak byla procházena a zpracovávána v LabVIEW a příklad výstupu byl již uveden na obrázku 3.3 v kapitole 3.3.

Mezi zajímavé poznatky, které měření ve vlaku přineslo, patří překvapivě vysoká přesnost detekce přejezdu přes výhybky, pokud je trať postavena tak, že dojde k odbočení z přímého směru. Ukázka přejezdu dvou výhybek po sobě, které vyvolaly změnu úhlu natočení o asi 20° jedním směrem a následně o stejný úhel zpět, je uvedena na obrázku 6.2. Graf vlevo zachycuje průběh úhlové rychlosti, ale zajímavější je graf vpravo, který zachycuje průběh úhlu ve stupních. Právě první dvě výchylky v průběhu úhlu představují přejezd přes jednu z výhybek zhlaví Smíchovského nádraží.



Obr. 6.2: Naměřený průběh úhlové rychlosti (vlevo) a vypočítaný průběh úhlu natočení (vpravo) během jízdy vlakem

Závěrem z měření ve vlaku tedy bylo především potvrzení možnosti detekce oblouků (úhlů otočení) a to s dostatečnou přesností. Dalším zjištěním bohužel byla naprostá nedostatečnost levného GNSS přijímače, který při umístění uvnitř vozu fungoval velmi špatně a GNSS data měla často výpadek.

6.2 Zaměření na tramvajový provoz

Po prvních zkušenostech s měřením v automobilu a následně i ve vlaku, výsledky sice vypadaly slibně, ale bylo zřejmé, že je nutné zařízení přepracovat do kompaktní verze, kterou bude možné jednoduše nainstalovat do vozidla a nechat dlouhodobě měřit bez nutnosti obsluhy. Tak vznikla druhá verze měřicího zařízení, která byla podrobně popsána v kapitole 4.2. Mechanická koncepce této verze byla od počátku zamýšlena pro použití v tramvajovém provozu, ale to nevylučuje nasazení i v jiných vozidlech. V rámci zaměření na tramvajový provoz pak byly aplikovány především tyto změny:

- Magnetické úchyty, které umožnili rychlou a snadnou montáž bez jakéhokoli zásahu do vozidla
- Díky oddělení měření a zpracování dat odpadla nutnost zahrnout do palubního zařízení PC
- Bateriové napájení měřicího zařízení umožnilo celodenní provoz bez připojování do sítě vozidla
- Externí anténa pro GNSS přijímač s magnetickým úchytem přinesla snadnou montáž i kvalitní příjem
- Robustní uzavíratelné pouzdro přístroje zajistilo nerušené měření a odolnost při manipulaci

6.2.1 Navázání spolupráce s PMDP

Po dokončení návrhu druhé verze měřicího zařízení, výrobě prototypu a naprogramování firmwaru, bylo možné začít s vlastním měřením. V rámci prvních testů bylo zařízení převáženo v rámci běžné přepravy v prostoru pro pasažéry na vybraných úsecích a výsledky byly analyzovány. Zde se ukázalo, že systém dokáže detekovat většinu segmentů tratě a že pro další vývoj bude nutné získat větší množství dat z více tras tak, aby bylo možné provést vyhodnocení.

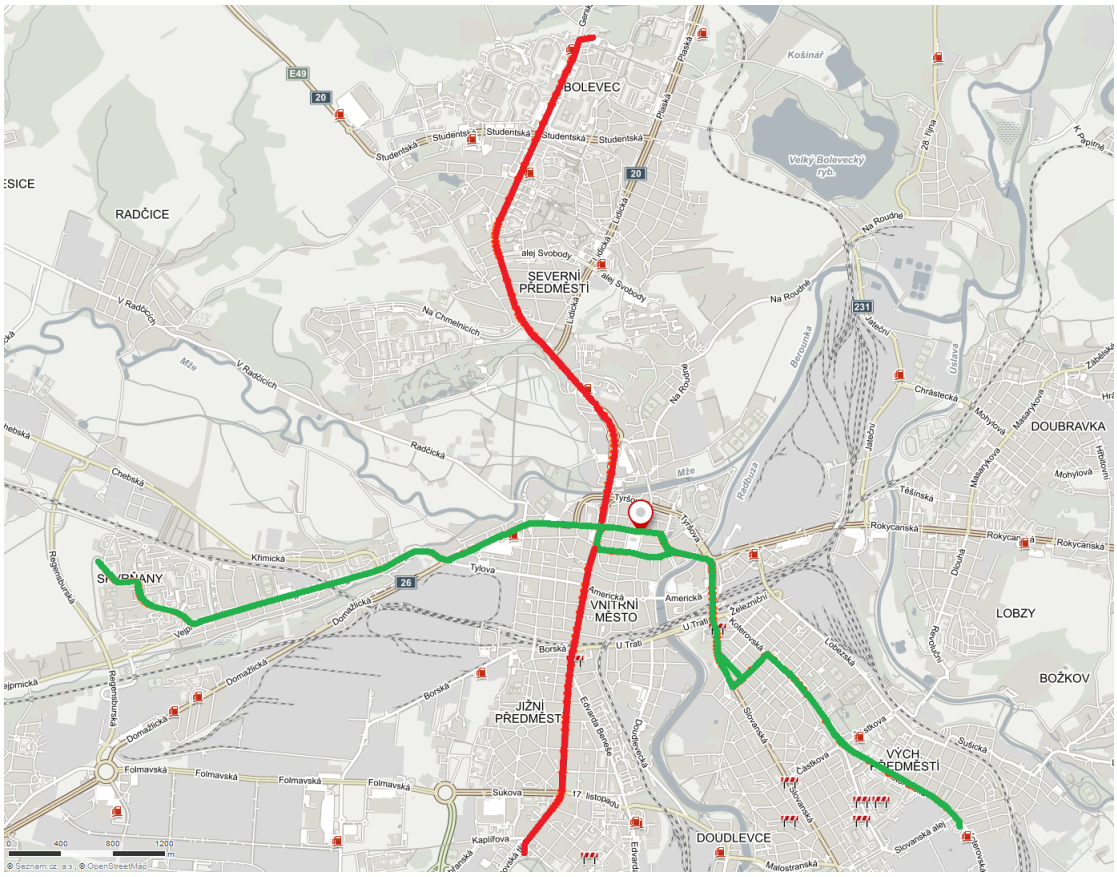
Možnost celodenních měření v tramvaji byla konzultována přímo s Plzeňskými městskými dopravními podniky (PMDP), resp. s vedoucím střediska elektrické dráhy - provoz. Díky vstřícnému jednání ze strany PMDP se následně podařilo domluvit první termín měření a měření realizovat. Jedním z klíčových faktorů, které umožnily realizaci byla právě kompaktnost a nezávislost měřicího zařízení, které nevyžadovalo žádnou obsluhu či zásahy do systémů vozidla. V opačném případě by byla realizace měření v této fázi projektu zbytečně nákladná nebo by nebyla z legislativních důvodů vůbec proveditelná.

V neposlední řadě pak domluva s PMDP přinesla možnost instalovat zařízení v uzamykatelném prostoru zadního stanoviště, takže zařízení mohlo nerušeně pracovat a bylo chráněno proti zásahům zvenčí, které by měření znehodnotili nebo dokonce proti odcizení.

6.2.2 Měření v tramvajovém provozu

Měření probíhalo na tramvajových linkách č. 2 a 4 v Plzni, jejichž trasy jsou vyznačeny v mapě na obrázku 6.3. Zelená trasa odpovídá lince č. 2 a červená pak lince č. 4. Z mapy je dobře vidět vzájemná odlišnost obou tras, kdy linka č. 2 je velmi členitá s oblouky s větším úhlem otočení, zatímco linka č. 4 obsahuje spíše rovné úseky a otáčení probíhá s menším úhlem. Právě tato odlišnost byla důvodem, proč pro měření byly zvoleny trasy 2 a 4, takže různý charakter tratí lépe otestuje navržený algoritmus.

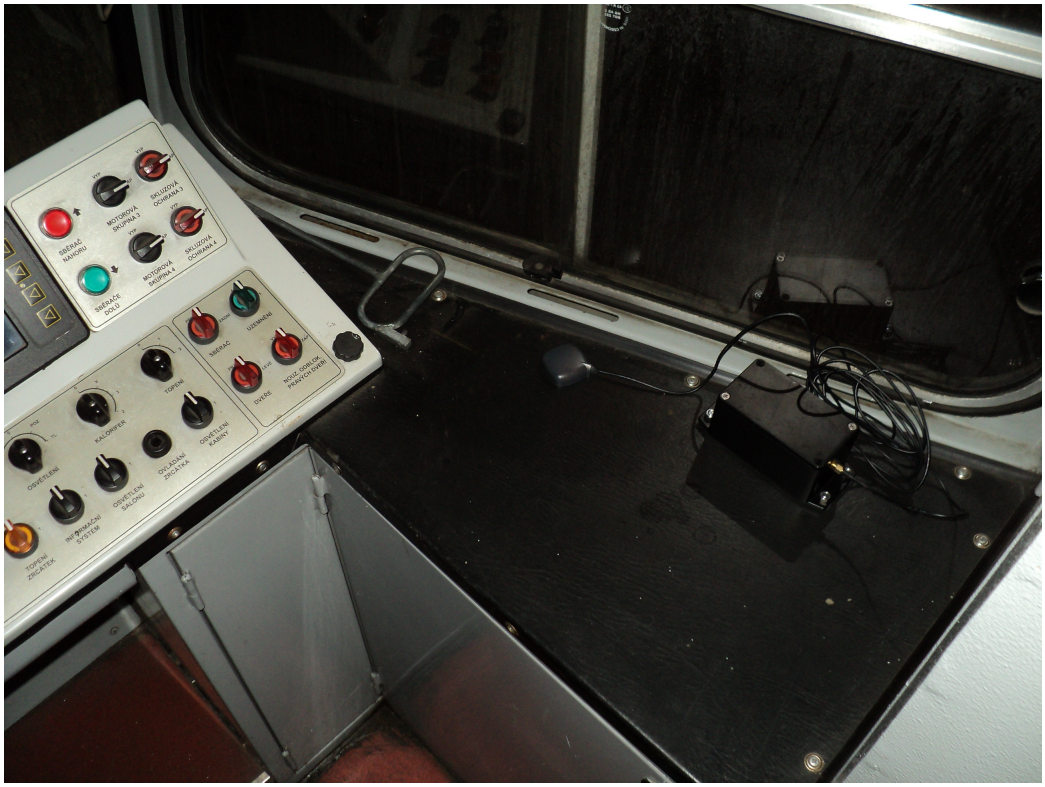
Aby bylo během dne nasbíráno co nejvíce dat (provedeno nejvíce jízd), byly pro měření v daný den, vytipovány vždy tramvaje se směnou od 4:00 do 20:00. Během dne se tedy v tramvaji vystřídaly dvě směny řidičů, což někdy znamenalo dva trochu odlišné styly jízdy, takže získaná data byla opět objektivnější. Počet jízd z konečné na konečnou na lince č. 2 vycházel na 13 v každém směru a na lince č. 4 pak 16 v jednom směru a 15 ve druhém. Součástí dne byly také přejezdy z depa na konečnou (ráno) a z konečné do depa (večer).



Obr. 6.3: Trasy tramvajových linek č. 2 (zelená) a 4 (červená)

Druhým kritériem pro výběr tramvaje byl její typ. Zde byl volen typ KT8D5-RN2P, neboť se jedná o obousměrnou tříčlánkovou tramvaj, tedy se stanovištěm řidiče na obou koncích. Toto kritérium nebylo sice nutnou podmínkou pro realizaci měření, ale uzavřením měřicího zařízení na nepoužívaném stanovišti, odpadla nutnost detailního instruování řidičů tak, aby například nedošlo k nechtěné manipulaci se zařízením apod. Na druhou stranu, toto řešení také řidiče nijak neomezuje a neovlivňuje jejich práci[1]. Ukázka instalace ve vozidle je uvedena na obrázku 6.4. Fotografie byla pořízena v ranních hodinách, těsně před odjezdem vozu z depa a měřicí zařízení je již zapnuté a uzavřené. Anténa GNSS přijímače je umístěna vedle pouzdra zařízení, co nejblíže k oknu. Bohužel, vzhledem k tomu, že anténa, koaxiální kabel i pouzdro zařízení jsou černé a boční plechový panel v tramvaji byl potažen černou koženkou, není viditelnost jednotlivých objektů na fotografii nejlepší.

[1]Lepší řešení z hlediska legislativy.



Obr. 6.4: Fotografie měřicího zařízení nainstalovaného v tramvaji

6.2.2.1 Ukázka výstupních dat

Výstupní data v grafické podobě byla již v podstatě ukázána na obrázku 5.5 v rámci popisu GUI vytvořeného navigačního programu v LabVIEW. Jelikož jsou data během výpočtu neustále k dispozici jako pole hodnot, je možné si na výstup do grafu nastavit libovolnou kombinaci veličin. Podobně je to i s textovým výstupem, který může obsahovat různé údaje. Pro zpracování výstupů navigačního systému za účelem validace a získání statistických údajů, byl textový výstup přesměrován do CSV souboru tak, aby jej bylo možné zpracovat jako spread sheet a zpracování tak částečně automatizovat. Na obrázku 6.5 je zachycen výřez z tabulky dat, která obsahuje výstupní data z navigačního systému, která byla načtena z CSV souboru. Je vidět, že data jsou již částečně zpracována a barevně odlišena. Například údaje týkající se detekce oblouků jsou označeny zeleně, ujetá vzdálenost modře a detekce zastavení a prostorů červeně. Data byla zpracována pro každé měření, přičemž každé měření obsahuje obvykle kolem 15 jízd z konečné na konečnou jedním směrem a zpravidla stejný počet jízd směrem opačným. Jízdy ve shodném směru byly zarovnány do sloupců tak, aby si detekované segmenty a ostatní údaje odpovídaly a bylo tak možné vytvářet statistiky. Na obrázku je pochopitelně zachycena jen velmi malá část dat a pro představu, celá tabulka obsahuje necelých deset tisíc buněk s navigačními údaji a další množství buněk se vzorci a zpracováním statistik. Takováto tabulka pak byla zpracována pro každé měření.

3.2.2017 14:24:03 Vozidlo zatočilo 10,666136° vlevo	3.2.2017 15:34:49 Vozidlo zatočilo 10,224253° vlevo
3.2.2017 14:24:03 Poloměr oblouku 605,669847 m	3.2.2017 15:34:49 Poloměr oblouku 537,051373 m
3.2.2017 14:24:04 Vozidlo opustilo prostor	3.2.2017 15:34:50 Vozidlo opustilo prostor
3.2.2017 14:24:35 Vozidlo v prostoru zastávky Škoda III, brána, Linka č. 2	3.2.2017 15:35:19 Vozidlo v prostoru zastávky Škoda III, brána, Linka č. 2
3.2.2017 14:24:37 Vozidlo zatočilo 58,403774° vpravo	3.2.2017 15:35:20 Vozidlo zatočilo 58,747314° vpravo
3.2.2017 14:24:37 Poloměr oblouku 244,058717 m	3.2.2017 15:35:20 Poloměr oblouku 228,791691 m
3.2.2017 14:24:43 Vozidlo zastavilo	3.2.2017 15:35:27 Vozidlo zastavilo
3.2.2017 14:24:43 Vozidlo ujelo 652,885724m	3.2.2017 15:35:27 Vozidlo ujelo 619,492043m
3.2.2017 14:24:45 Vozidlo se rozjelo	3.2.2017 15:35:28 Vozidlo opustilo prostor
3.2.2017 14:24:50 Vozidlo zastavilo	3.2.2017 15:35:30 Vozidlo v prostoru zastávky Škoda III, brána, Linka č. 2
3.2.2017 14:24:50 Vozidlo ujelo 1,377837m	3.2.2017 15:35:32 Vozidlo se rozjelo
3.2.2017 14:24:54 Vozidlo se rozjelo	
3.2.2017 14:25:04 Vozidlo opustilo prostor	3.2.2017 15:35:43 Vozidlo opustilo prostor
3.2.2017 14:25:12 Vozidlo zatočilo 53,210877° vlevo	3.2.2017 15:35:51 Vozidlo zatočilo 52,559611° vlevo
3.2.2017 14:25:12 Poloměr oblouku 46,272555 m	3.2.2017 15:35:51 Poloměr oblouku 56,190584 m
3.2.2017 14:26:01 Vozidlo v prostoru zastávky CAN Skvrňanská, Linka č. 2	3.2.2017 15:36:35 Vozidlo v prostoru zastávky CAN Skvrňanská, Linka č. 2
3.2.2017 14:26:18 Vozidlo zastavilo	3.2.2017 15:36:41 Vozidlo zastavilo
3.2.2017 14:26:18 Vozidlo ujelo 310,161844m (přičteno 1,377837m)	3.2.2017 15:36:41 Vozidlo ujelo 292,790936m
3.2.2017 14:26:21 Vozidlo se rozjelo	3.2.2017 15:37:01 Vozidlo se rozjelo
3.2.2017 14:26:24 Vozidlo zastavilo	3.2.2017 15:37:04 Vozidlo zastavilo
3.2.2017 14:26:27 Vozidlo se rozjelo	3.2.2017 15:37:07 Vozidlo se rozjelo
3.2.2017 14:26:36 Vozidlo opustilo prostor	3.2.2017 15:37:15 Vozidlo opustilo prostor
3.2.2017 14:27:05 Vozidlo zatočilo 27,751229° vpravo	3.2.2017 15:37:58 Vozidlo zatočilo 27,840824° vpravo
3.2.2017 14:27:06 Poloměr oblouku 324,396816 m	3.2.2017 15:37:59 Poloměr oblouku 360,196765 m
3.2.2017 14:27:20 Vozidlo v prostoru zastávky Palackého náměstí, Linka č. 2	3.2.2017 15:38:17 Vozidlo v prostoru zastávky Palackého náměstí, Linka č. 2
3.2.2017 14:27:35 Vozidlo zastavilo	3.2.2017 15:38:38 Vozidlo zastavilo
3.2.2017 14:27:35 Vozidlo ujelo 728,060520m	3.2.2017 15:38:38 Vozidlo ujelo 873,293590m
3.2.2017 14:27:37 Vozidlo se rozjelo	3.2.2017 15:38:45 Vozidlo se rozjelo
3.2.2017 14:27:39 Vozidlo zastavilo	
3.2.2017 14:27:41 Vozidlo se rozjelo	
3.2.2017 14:27:44 Vozidlo zastavilo	
3.2.2017 14:27:46 Vozidlo se rozjelo	3.2.2017 15:38:46 Vozidlo opustilo prostor
	3.2.2017 15:38:52 Vozidlo v prostoru zastávky Palackého náměstí, Linka č. 2
3.2.2017 14:27:56 Vozidlo opustilo prostor	3.2.2017 15:38:55 Vozidlo opustilo prostor

Obr. 6.5: Ukázka z analýzy výstupních dat navigačního systému

6.2.3 Dosažené parametry navigačního systému

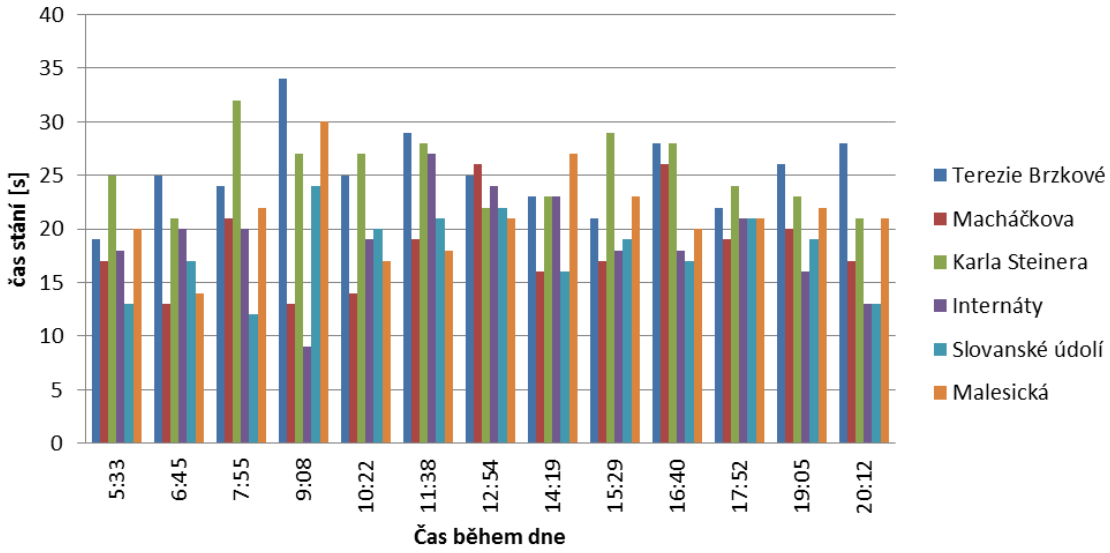
Po zpracování všech naměřených dat výše popsaným způsobem, již bylo možné začít vyhodnocovat určité vlastnosti navrženého navigačního systému a zpracovávat pro ně statistiku. Parametry pro vlastní validaci systému byly vybrány jednak na základě měřených, resp. počítaných veličin, ale také s ohledem na princip identifikace charakteristických segmentů trati. Tyto jednotlivé parametry, jejich odchylka a průběh budou předmětem několika následujících podkapitol.

6.2.3.1 Úspěšnost v detekování zastávek

Detekce zastávek, respektive zastavení za účelem nástupu a výstupu cestujících, využívá v současné verzi systému jak data z IMU tak z GNSS. Každé zastavení vozidla je totiž detekováno systémem na základě zpracovaných inerciálních veličin. Pokud stání trvá déle než nastavená mez, kontroluje algoritmus, zda se vozidlo také nachází v některém z prostorů, definovaném v databázovém souboru. Pokud jsou obě podmínky splněny, je detekováno stání v zastávce.

Konec stání v zastávce je pak podmíněn uvedením vozidla znovu do pohybu a opuštěním prostoru. Odečtením časových značek obou událostí je tedy možné určit čas strávený

v zastávce. Ačkoli toto řešení nereflektuje přímo čas pro nástup a výstup cestujících[2], pro ukázku možností navrženého systému postačuje. Využití těchto časových údajů pak může být zajímavé například pro provozovatele hromadné přepravy osob. Příklad výstupu této části algoritmu je uveden v diagramu na obrázku 6.6. Zde je zachycen vývoj času stráveného v několika po sobě jdoucích zastávkách během dne.



Obr. 6.6: Diagram vývoje času stráveného v zastávkách

Co se týká vlastní úspěšnosti detekce zastávek, lze problém rozdělit na dvě části. Jedna z nich je určení přítomnosti v prostoru z GNSS dat, která je podmíněna kvalitou GNSS příjmu. Zde občas nastával problém, že GNSS data byla příliš zpožděná a k zastavení vozidla tak došlo dříve než k detekci prostoru. Tato chyba nastala asi ve 3% projetých zastávek během všech měření.

Druhá část problému detekce se týká inerciálních dat a výpadek GNSS ještě nemusí nutně znamenat nedetekování zastávky. Pokud by detekce byla založena pouze na inerciálních datech, musel by algoritmus vyhodnotit jednak délku stání, ale také „polohu stání“ vůči předem známému sledu jednotlivých segmentů tratě. Například, pokud se zastávka nachází mezi dvěma segmenty definovanými úhlem a poloměrem, pak je algoritmus schopen určit zda se také navigované vozidlo nachází v tomto úseku (první segment už projelo a druhý ještě ne). Dojde-li tedy k zastavení, může se jednat o stání v zastávce. K dalšímu zpřesnění pak lze zapojit ještě vypočítanou ujetou vzdálenost porovnávanou se skutečnou vzdáleností od prvního segmentu k zastávce v rámci definované tolerance[3] apod. Pro tuto metodu je tedy nutné spolehlivě detekovat vlastní zastavení vozidla. Algoritmus, který řeší detekci zastavení, byl podrobně popsán výše a spolehlivost této části systému je velmi vysoká. Během všech provedených měření došlo k nedetekování zastavení pouze ve dvou

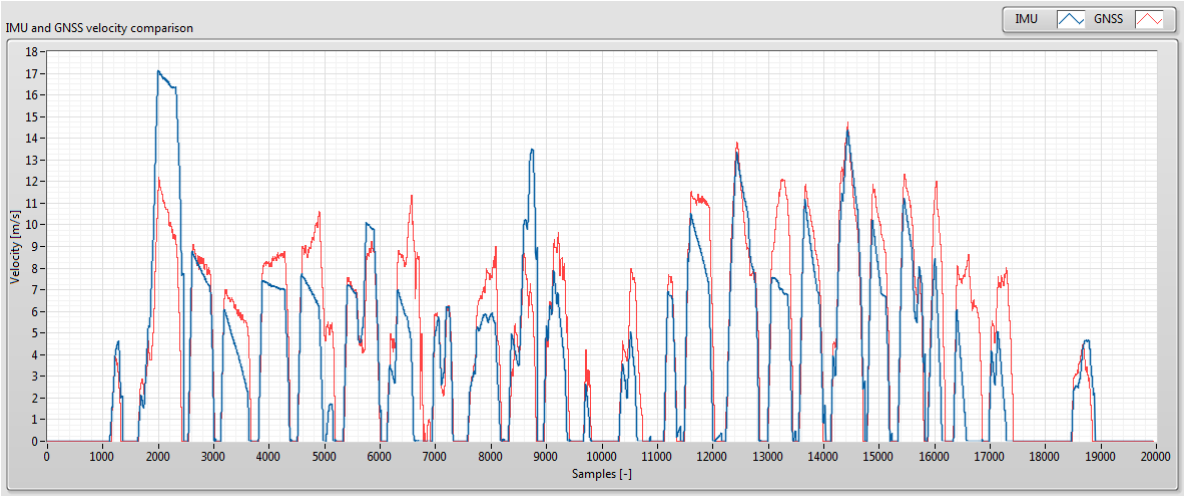
[2]Bylo by třeba do systému zavést další informaci, například o otevření dveří

[3]Měření vzdálenosti není příliš přesné.

případech a pokud zde vozidlo skutečně zastavilo, byla by chyba vyjádřená v procentech menší než 0,1%.

6.2.3.2 Porovnání vypočítané rychlosti s GNSS

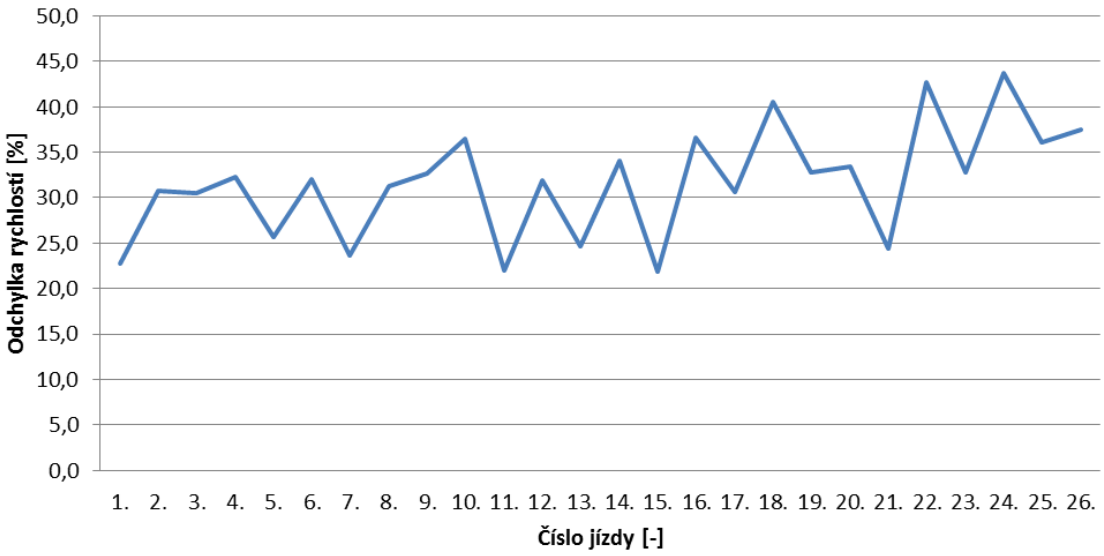
Klíčovou veličinou vypočítanou z inerciálních dat, resp. z měřeného zrychlení, je aktuální rychlost vozidla. Rychlost totiž vstupuje jako parametr do dalších algoritmů, např. pro výpočet ujeté vzdálenosti nebo pro výpočet poloměru oblouku, a její přesnost tak do značné míry určuje celkovou chybu uvedených výpočtů. V rámci testování popisovaného navigačního systému je pro výpočty používána výhradně rychlost spočítaná z inerciálních dat. Současně s touto rychlostí je do souboru naměřených dat také ukládána rychlost vůči Zemi, která je obsažena v datech z GNSS přijímače. Kvalitu výpočtu rychlosti je tedy možné validovat na základě porovnání obou rychlostí. Pro představu jak se vypočítaná a GNSS rychlost od sebe liší, je na obrázku 6.7 zachycena část čelního panelu jednoho z testovacích VI, kde jsou obě rychlosti ukázány v grafu. Modře je znázorněna počítaná rychlost a červeně pak rychlost čtená z GNSS přijímače. Graf zachycuje typický průběh rychlosti během jízdy vozidla z jedné konečné na druhou, kde krátké úseky s nulovou hodnotou představují zastávky a jednotlivé vrcholy pak přejezdy mezi nimi. Z průběhů rychlosti mezi zastávkami je také dobře patrný styl jízdy, kdy prudký nárůst je dán akcelerací po rozjezdu, pak následuje úsek po který je rychlost udržována na určité hodnotě a nakonec je brzděním způsoben rychlý pokles rychlosti až k nule.



Obr. 6.7: Porovnání vypočítané rychlosti (modrá) a rychlosti získané z GNSS (červená)

V rámci porovnání obou rychlostí je v grafu na obrázku 6.8 uveden průběh procentuální odchylky vypočítané rychlosti vzhledem k rychlosti z GNSS během jednoho měřicího dne, kde čísla na vodorovné ose představují pořadová čísla jednotlivých jízd. Průměrná hodnota odchylky pro všechna měření vychází kolem 30%. Toto číslo ale není zcela přesné, resp. nemá takovou vypovídající hodnotu, protože porovnání obou signálů probíhalo vzorek po vzorku, což nerespektuje mírný vzájemný fázový posun obou průběhů, který je částečně

vidět v grafu na obrázku 6.8. Vypočítaná rychlost má totiž díky algoritmu mírné zpoždění oproti rychlosti čtené z GNSS a přesnější informaci o rozdílu obou signálů by bylo nutné získat složitěji, např. korelací. Takto vychází chyba mírně vyšší než ve skutečnosti je, ale pro představu o přesnosti výpočtu je tato informace dostačující.



Obr. 6.8: Průběh odchylky vypočítané rychlosti od rychlosti z GNSS

6.2.3.3 Přesnost ve vypočítané vzdálenosti

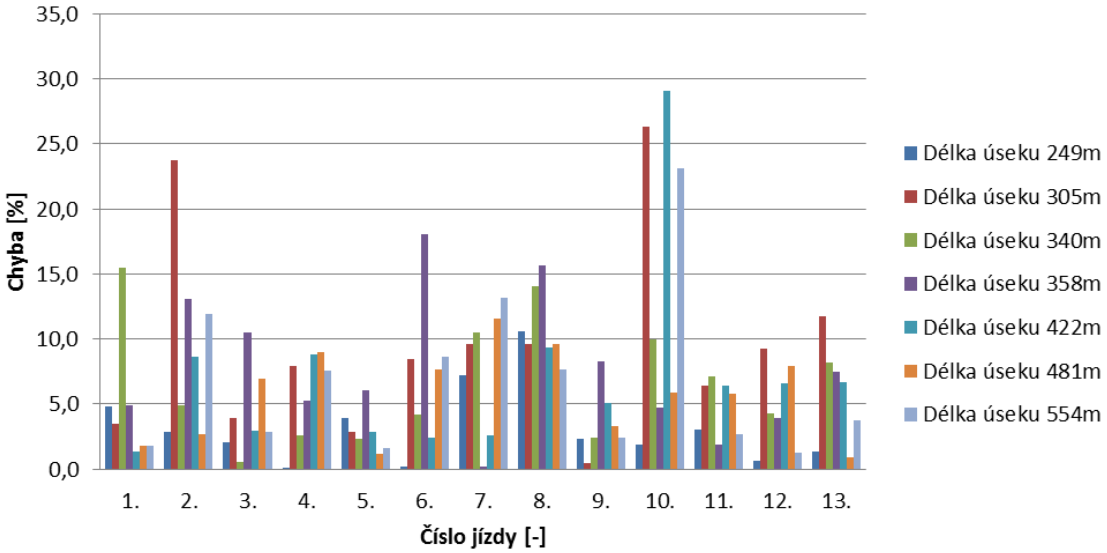
Výpočet ujeté vzdálenosti je realizován integrací rychlosti vozidla v čase a za předpokladu použití pouze inerciálních dat, je tato rychlost získávána integrací dopředné složky zrychlení. Toto nastavení bylo použito pro všechno provedené zpracování dat tak, aby byla ověřena přesnost systému ve stavu, kdy GNSS data nejsou dostupná.

Z výše uvedeného plyne, že jakákoli chyba v měření a výpočtu zrychlení se projeví v ujeté vzdálenosti s kvadrátem a neustále se k výsledku přičítá. Z tohoto důvodu nebyla od výpočtu vzdálenosti očekávána vysoká přesnost, ale i tak musí navigační systém úseky tratě bezpečně rozpoznat. Detekce fungovala během všech měření správně a prakticky všechny úseky mezi zastávkami na projížděných trasách byly bezpečně rozpoznány s úspěšností detekce vyšší než 99%.

Co se týká přesnosti výpočtu délky úseku, pak průměrná chyba vůči nominální hodnotě změřené v mapě, dosahovala 38%. Jiný způsob získání nominální délky úseku, je využít vlastní měřicí systém a opakovaným měřením stejného úseku stanovit průměrnou naměřenou hodnotu. Tato hodnota sice nemusí odpovídat reálné vzdálenosti, ale její vypovídající hodnota v rámci detekce segmentů je patrně vyšší. Jelikož byly stejné trasy během měření projety vícenásobně, byla vyzkoušena i tato metoda. Průměrná chyba vůči takto získané referenční hodnotě pak dosahovala jen 15%.

Kromě průměrné chyby, která vychází poměrně dobře, je ale důležitější, jakých maximálních hodnot chyba dosahuje během opakovaných průjezdů stejného úseku. Tato informace pak určuje, s jakou opakovatelnou přesností dokáže algoritmus identifikovat úsek tratě o definované délce. Průběh chyby pro několik opakovaně projížděných úseků je ukázán na diagramu na obrázku 6.9. Z obrázku je vidět, že se chyba většinou pohybuje do uvedených 15% a v extrémních případech dosahuje až 30%.

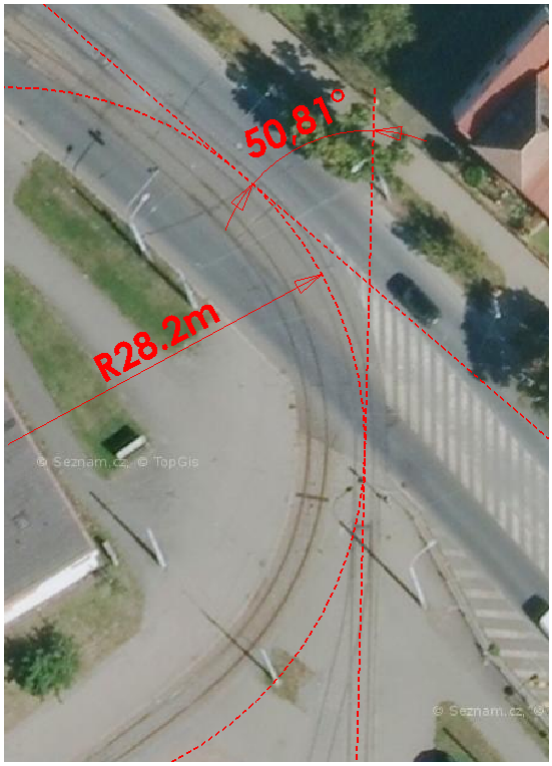
Z principu integrace také plyne, že chyba měření vzdálenosti je ještě ovlivněna délkou úseku, resp. časem po který vozidlo úsek projíždí. Pro srovnání je tedy vhodné ještě uvést, že nominální (skutečná) délka nejkratších úseků se pohybovala kolem 200 m a nejdelší vzdálenost pak kolem 800 m.



Obr. 6.9: Fluktuace chyby výpočtu vzdálenosti během několika jízd

6.2.3.4 Dosažená přesnost ve výpočtu úhlu

V rámci validace algoritmů pro výpočet úhlu a poloměru projížděného oblouku, bylo nutné získat nejprve reálné referenční hodnoty, se kterými je možné porovnat výstupní data. Referenční hodnota úhlu otočení, směr otočení a referenční poloměr oblouku byly získány ručně, měřením v mapě, resp. v leteckých snímcích tratě a přilehlého okolí. Způsob měření je ukázán na obrázku 6.10. V CAD softwaru byly vždy přes snímek části trati nakresleny dvě přímky procházející osou kolejového svršku a byl změřen úhel, který spolu v místě oblouku svírají. Dále byla do oblouku aproximována kružnice tak, aby její část byla totožná s osou kolejnic v místě oblouku. Poloměr této kružnice pak představuje poloměr oblouku a ze znalosti měřítka snímku (mapy) je možné spočítat skutečnou hodnotu v metrech. Tímto způsobem byla zmapována každá testovaná trasa v obou směrech a výsledkem je sekvence oblouků s definovanými parametry, která je v podstatě unikátní pro každou trasu a směr.



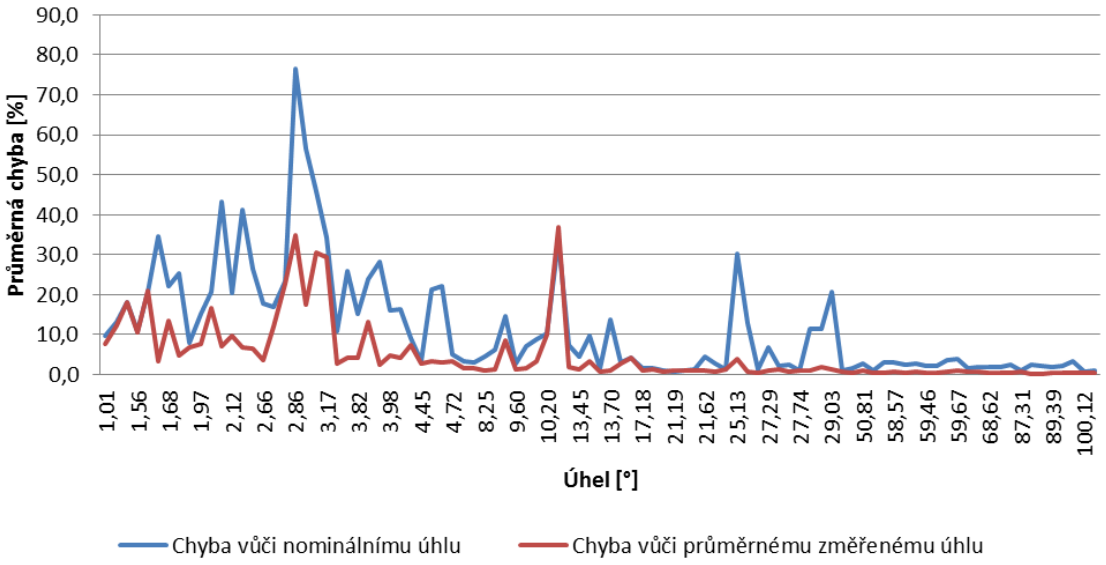
Obr. 6.10: Odměřování skutečných parametrů segmentů tratě

Ohledně vypočítaného úhlu otočení byla určena procentuální chyba vůči hodnotám získaným výše popsaným způsobem. Tato chyba v průměru vychází přibližně 12%, což je zbytečně velká hodnota. Lze zde totiž aplikovat stejný postup jaký byl popsán v případě výpočtu ujeté vzdálenosti a sice, jako referenční hodnoty úhlu použít průměrované hodnoty změřené přímo navigačním systémem. V tomto případě chyba v průměru vychází pouze 5% a vzhledem k tomu, že ruční měření v mapě má přesnost jen asi $\pm 2°$, je referenční hodnota úhlu změřená navigačním systémem často blíže k realitě, zvlášť v případě malých úhlů.

Velikost úhlu hraje v jeho detekci a výpočtu významnou roli a dosažená přesnost algoritmu na ni závisí, jak je vidět z grafu na obrázku 6.11. Je patrné, že tato závislost platí jak pro chybu vztaženou k hodnotě změřené v mapě (modrý průběh), tak i pro chybu vztaženou k průměrné hodnotě změřené vlastním systémem (červený průběh). Z grafu je také vidět, že v extrémním případě chyba dosahuje více než 35% a je výrazně vyšší pro malé úhly. Pokud bychom tedy detekované oblouky omezili na min. úhel 5°, byla by průměrná chyba jen 2% a pro úhly větší než 45° pak pouze 0,5%.

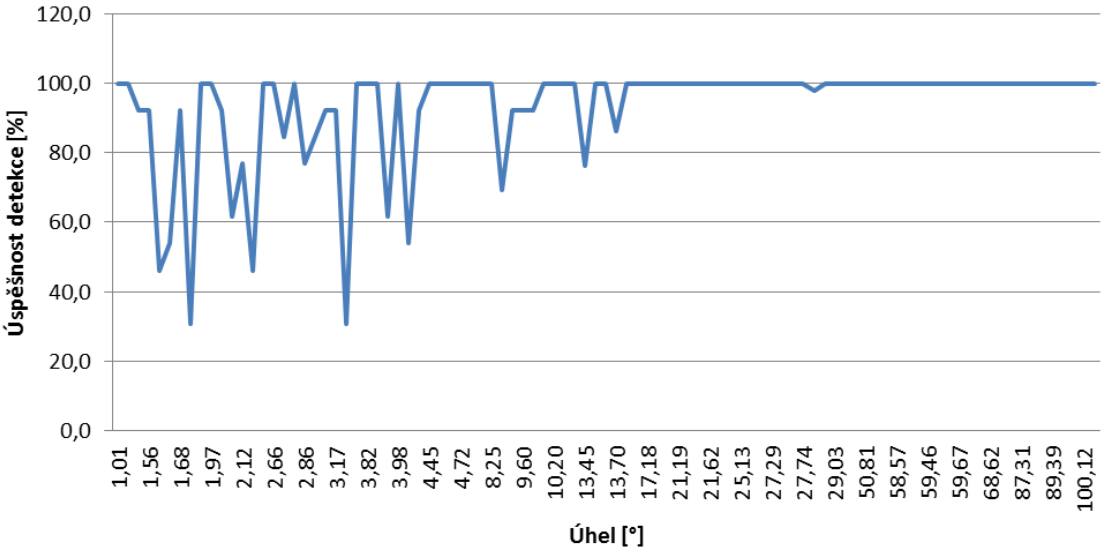
Pokud je změna úhlu během jízdy natolik nevýrazná, že prakticky zmizí v rušivých složkách obsažených v inerciálních datech, systém nedokáže úhel spočítat a projetý segment tak není detekován. Tato vlastnost nemusí však příliš vadit, protože k těmto nevýrazným změnám dochází pouze při velmi malých úhlech (nepatrná změna směru na jinak rovném úseku apod.) pohybujících se v hodnotách do 3°. Detailnější přehled o úspěšnosti detekce úhlů v závislosti na jejich velikosti je poskytnut v grafu na obrázku 6.12. Je vidět,

že pro úhly od cca 10° je úspěšnost detekování už prakticky 100%. Průměrná úspěšnost detekování segmentu bez ohledu na hodnotu úhlu pak vychází přes 92%.



Obr. 6.11: Závislost chyby výpočtu úhlu na velikosti úhlu

Z hlediska stability detekce a výpočtu úhlu v rámci stejného segmentu je zajímavá hodnota směrodatné odchylky, spočítaná ze série hodnot změřených při opakovaném projíždění stejného oblouku. Tato hodnota se pohybuje v rozmezí 0,07° až 2,67° v rámci všech testovaných úhlů, což znamená, že úhly od 3° výše by měly být systémem bezpečně rozlišitelné.

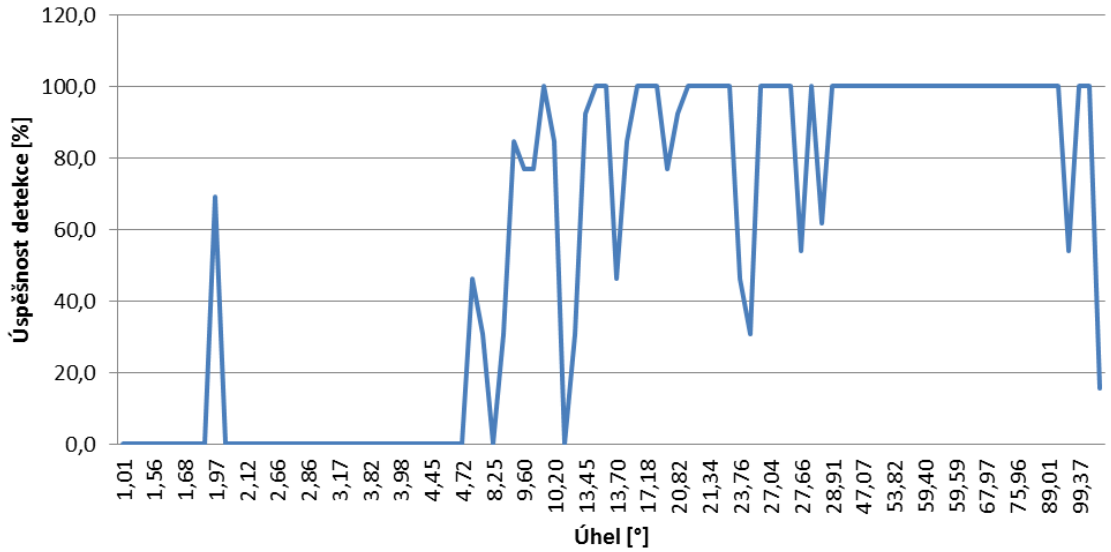


Obr. 6.12: Závislost úspěšnosti detekce otočení na velikosti úhlu

6.2.3.5 Dosažená přesnost ve výpočtu poloměru

Algoritmus pro výpočet poloměru oblouku, který byl v této práci popsán, je asi nejkomplikovanějším článkem celého navigačního systému a vykazuje největší citlivost na chyby ve vstupních datech. Poloměry oblouků získané z leteckých snímků výše popsaným způsobem jsou značně nepřesné a pro některé malé úhly je není možné vůbec určit. Hodnoty, které se změřit podařilo se pohybují v rozmezí 20 až 700 m. Takto velký rozptyl a nepřesnost referenčních hodnot jsou jednou z hlavních příčin, proč chyba výpočtu vychází v průměru téměř 45%. Aplikace průměrných změřených hodnot jako reference situaci sice zlepší, ale i přesto chyba dosahuje v průměru 26%.

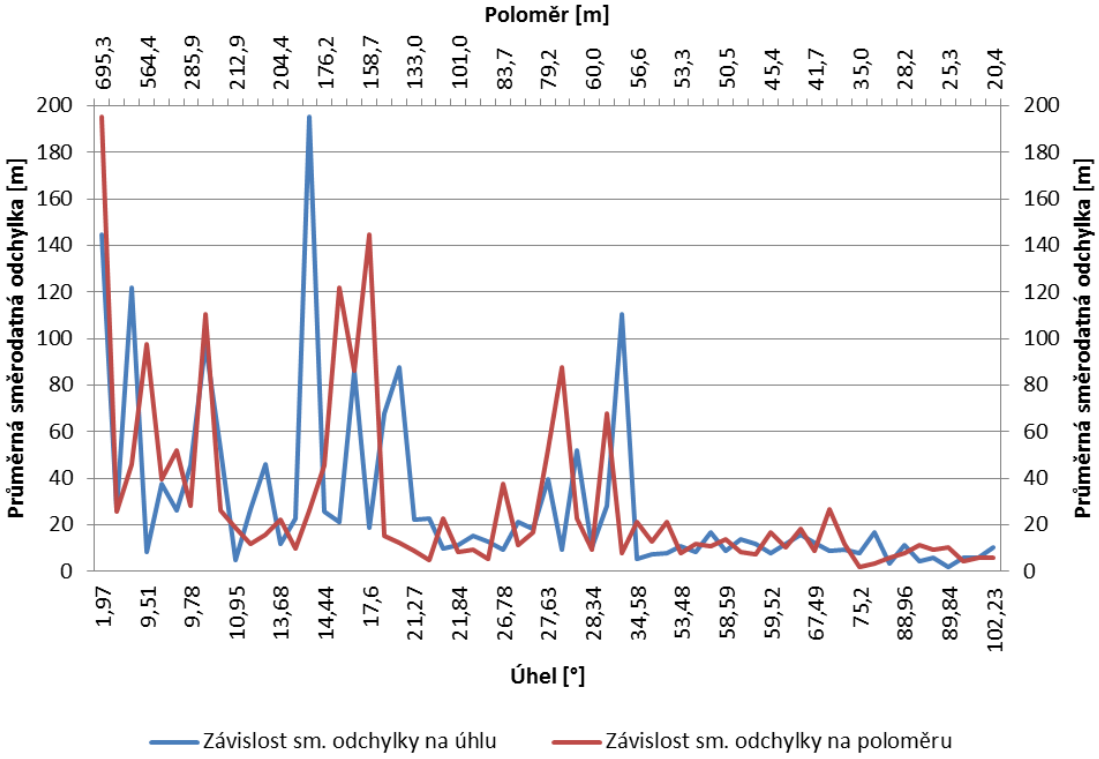
Další problém nastává v případě malých úhlů, kdy systém poloměr nedokáže většinou určit a pokud ano, vrátí velmi chybnou hodnotu. Úspěšnost provedení výpočtu poloměru je tedy závislá především na úhlu otočení, jak je to patrné z grafu na obrázku 6.13. Je vidět, že zcela spolehlivě dokáže algoritmus spočítat poloměr pro oblouky s úhlem otočení v rozmezí 30° až 90°. Tato informace je zásadní, protože po omezení úhlů na uvedený interval, klesne průměrná chyba výpočtu (vztažená k průměrované hodnotě) na 14% a především její maximální hodnota nepřekračuje 23%. Tato přesnost je již pro identifikaci segmentu (oblouku) na základě definovaného úhlu a poloměru postačující. Pro oblouky mimo uvedený interval úhlů, však zatím lépe vychází použít pro identifikaci pouze úhel.



Obr. 6.13: Závislost úspěšnosti výpočtu poloměru na velikosti úhlu

Stejně jako v případě úhlu, platí i pro výpočet poloměru, že stabilita detekce a výpočtu v rámci stejného segmentu je určena směrodatnou odchylkou a rozlišení segmentů s poloměrem menším než její hodnota nelze zaručit. V případě poloměru nezávisí směrodatná odchylka pouze na velikosti úhlu, ale je závislá také na samotném poloměru. Obě závislosti jsou zachyceny v grafu na obrázku 6.14. Je vidět, že směrodatná odchylka je

menší než 20 m pro úhly větší než asi 35° a pro poloměry menší než 60 m. V případě úhlu, tato závislost také potvrzuje interval úspěšné detekce, který byl uveden výše.



Obr. 6.14: Závislost průměrné směrodatné odchylky poloměru na úhlu a poloměru

7

Závěr

Byl navržen, otestován a zhodnocen navigační systém, jehož funkce je založena na principu detekce charakteristických segmentů tratě. Díky tomuto principu je systém určen primárně pro použití u kolejových vozidel nebo obecně u vozidel s předem známou geometrií trasy, a proto měření a testování systému probíhalo, s ohledem na možnou reálnou aplikaci, v tramvajovém provozu. Princip detekce segmentů ale především umožňuje dlouhodobé fungování navigace i bez příjmu údajů o poloze z GNSS přijímače, což tento systém odlišuje od běžných navigačních přístrojů.

Navržený systém je rozdělen na dvě části, z nichž první představuje zařízení pro sběr inerciálních dat, které se instaluje do vozidla a druhá část je pak více méně softwarová a představuje program zpracovávající měřená data. Ačkoli jsou obě části rozděleny a navigační algoritmus zpracovává data až po skončení měření a mimo vozidlo, chová se software k souboru naměřených dat tak, jako by měření právě probíhalo a zpracovává data vzorek po vzorku. V podstatě lze říci, že kdyby se do vozidla společně s měřicím zařízením umístil i počítač se softwarovou částí, bylo by na monitoru možné sledovat výstupy v reálném čase. Řešení, kdy jsou obě části oddělené, je však pro vývoj praktičtější a umožňuje testovat algoritmy po různých úpravách znovu nad stejnými daty, bez nutnosti nového měření. Na druhou stranu, on-line koncepce navržených algoritmů zaručuje, že implementace systému v reálném čase je zaručena.

Vlastní měřicí zařízení využívá MEMS senzory, které měří zrychlení a úhlovou rychlost a bylo v rámci práce realizováno ve dvou prototypech. Druhá verze zařízení byla konstruována jako kompaktní uzavřený modul s vlastním bateriovým napájením a s možností nezávislého provozu po několik desítek hodin. Součástí druhé verze je i kvalitní GNSS přijímač, který je doplněn externí anténou. Softwarová část byla kompletně tvořena v prostředí LabVIEW a implementuje v sobě geometrické a navigační výpočty tak, jak byly popsány v teoretické části této práce.

Poslední část shrnutí by měla být věnována dosaženým parametrům, které byly podrobně popsány v předchozí kapitole. Zde je nutné uvést, že tyto parametry platí pro režim, kdy pro navigační výpočty jsou používána pouze inerciální data, čili je trvale navozen stav bez příjmu GNSS. Dosažené parametry systému jsou uvedeny v tabulce 7.1.

Tab. 7.1: Parametry navrženého navigačního systému

Parametr	**Hodnota**
Chyba v detekci zastavení vozidla	Méně než 0,1%
Odchylka vypočítané rychlosti od reálné (z GNSS)	30% průměrně
Chyba výpočtu ujeté vzdálenosti	30% maximálně [a)]
Chyba výpočtu úhlu otočení (úhly od 5° výše)	2% průměrně (35% max.)
Úspěšnost detekce odbočení (úhly od 10° výše)	Více než 99%
Chyba výpočtu poloměru oblouku (úhly od 35° a poloměry do 60 m)	14% průměrně (23% max.)

[a)] Hodnota je platná pro úseky v délce od 200 m do 800 m bez GNSS.

Z výše uvedeného shrnutí plyne, že cíle vytyčené na začátku práce byly splněny prakticky beze zbytku a bylo vytvořeno jak fyzické měřicí zařízení instalovatelné do vozidla, tak i software pro zpracování naměřených dat, který implementuje vlastní navigační algoritmy. V rámci výběru metod vhodných pro kolejová vozidla byl navržen speciální algoritmus pro určování poloměru projetého oblouku, který byl následně implementován a společně s dalšími navigačními algoritmy testován. Měření reálných dat bylo prováděno v tramvajovém provozu a výsledky byly statisticky zpracovány za účelem ověření dosažených parametrů navigačního systému.

Hlavní přínos navrženého systému spočívá především v odlišném přístupu k navigační úloze, než je tomu u tradičních systémů založených pouze na GNSS a prediktivní filtraci. Popsaný přístup vychází z nezávislé detekce projížděných charakteristických segmentů tratě, jako jsou oblouky, výhybky apod. a jejich identifikaci a porovnání s databází. Systém je tak schopen spolehlivě určit svoji polohu v rámci dané trati při každé změně segmentu a to bez nutnosti přístupu k GNSS datům. Na druhou stranu, pokud absence GNSS dat trvá i nadále v rámci jednoho dlouhého segmentu, je díky fenoménu integrační chyby po určitém čase nepřesnost absolutní polohy velmi velká. Dojde-li ale k opětovné změně segmentu, systém znovu přesně určí svoji polohu v rámci definované trasy a integrační chyba je tak vynulována. Pokud tedy například v tunelu nebo na jiném místě bez možnosti příjmu GNSS signálu vozidlo přejede výhybku či započne nebo ukončí průjezd obloukem, dokáže navržený systém znovu určit přesně polohu, zatímco běžný navigační systém v takovém případě dále čeká na obnovení příjmu GNSS a chyba estimované polohy tak neustále kvadraticky roste.

Dalším přínosem navrženého systému je také vhodný kompromis mezi nákladností a dosaženým výkonem. Za pomoci relativně levných senzorů, které jsou spíše určené pro mobilní zařízení a trvalou součinnost s GNSS přijímači, bylo dosaženo přijatelné přesnosti

i bez syntézy s GNSS daty.

Bylo by pochopitelně možné navrhnout systém s výkonnými inerciálními senzory, několika GNSS přijímači, napojením na systémy vozidla a dosáhnout tak řádově lepších výsledků. Na druhou stranu, takové zařízení bude jen těžko natolik kompaktní jako popsaný systém a z ekonomického hlediska nesrovnatelně nákladnější a právě ekonomické hledisko hraje v dnešní době velmi významnou roli. Pro menší společnosti, které se zaměřují na informační, dispečerské nebo monitorovací systémy pro veřejnou dopravu, by pak navržené zařízení mohlo být zajímavé již v této prototypové verzi a lze si tak představit kompaktní, ekonomicky přijatelný subsystém fungující jako palubní část složitějšího celku, jejíž výstup bude odesílán na dispečink již existující vysílací sítí.

Určení, respektive možné aplikace popsaného navigačního systému tedy směřuje primárně k dispečerským a dohledovým a monitorovacím systémům. Přínosem pro provozovatele dopravy by pak mohlo být také získávání některých statistických údajů, jakým je například vývoj doby stání ve stanici a s tím spojený obrat cestujících nebo sledování průjezdů taktovacími body v rámci kontroly a plánování. Jiné využití měřicího systému by ještě mohlo být nasazení jako záznamové zařízení. Hustota záznamu měřených dat by pravděpodobně vždy byla vyšší než u dat přenášených na operátorské stanoviště. V případě potřeby by pak data uložená na záznamovém médiu zařízení, mohla posloužit k detailnímu rozboru určité jízdní situace.

Aplikací, která je momentálně velmi aktuální, je nasazení systému jako monitorovacího zařízení v rámci výzkumu z oblasti elektromobility. V současné době probíhá ve spolupráci s PMDP výzkum v oblasti hybridních bateriových trolejbusů, kde je cílem zjistit, jakým způsobem může být na dané trase vozidlo provozováno tak, aby bylo možné odhadnout dojezd pro části trasy, kde je možný pouze provoz na baterii. S tím souvisí také vliv na trolejovou síť a přenosovou soustavu, protože je nutné odhadnout, kdy a kde které vozidlo bude dobíjet baterie tak, aby nedošlo k ovlivnění běžného provozu ostatních vozidel. V rámci elektromobility se nabízí možnost spojit inerciální údaje z navigačního systému s daty o energetických poměrech v pohonné soustavě vozidla, a tak získat například představu o tom, jak souvisí efektivita rekuperace elektrické energie se stylem jízdy (akcelerace a brzdění), dopravní situací nebo projížděným terénem.

Bez ohledu na aplikaci navrženého systému platí, že určování polohy za pomoci detekce charakteristických segmentů je v principu možné a v praxi realizovatelné a dosažené výsledky popsané výše to dostatečně potvrzují. Dalším vývojem, testováním a implementací nových poznatků je pak možné navigaci ještě více zpřesňovat a otevřít tak cestu k dalším aplikacím a komerčním partnerům.

Literatura

[1] DANA, Peter H. *Global Positioning System Overview* [online]. Department of Geography, University of Texas at Austin, 1994 [cit. 2016-01-20]. Dostupné z: http://www.colorado.edu/

[2] *GPS.gov* [online]. USA: U.S. Air Force, 2016 [cit. 2016-01-20]. Dostupné z: http://www.gps.gov/

[3] Síť permanentních stanic GNSS České Republiky. *Zeměměřičský úřad.* [online]. Praha: Zeměměřičský úřad, 2016 [cit. 2016-01-20]. Dostupné z: http://czepos.cuzk.cz/

[4] *Information-analytical centre* [online]. Korolyov (Russia): Federal space agency, 2016 [cit. 2016-01-20]. Dostupné z: https://www.glonass-iac.ru/en/

[5] *European Space Agency* [online]. Paris: European Space Agency, 2016 [cit. 2016-01-20]. Dostupné z: http://m.esa.int/ESA

[6] BeiDou Navigation Satellite System. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (US-CA): Wikimedia Foundation, 2001- [cit. 2016-01-20]. Dostupné z: https://en.wikipedia.org/

[7] NMEA data. *GPS Information.net: GPS Information Resource* [online]. 2013 [cit. 2016-01-20]. Dostupné z: http://www.gpsinformation.org/dale/nmea.htm

[8] Sanso, Fernando, ed. a Sideris, Michael G., ed. *Geoid determination: theory and methods*. Berlin: Springer, 2013. xxi, 734 s. Lecture notes in earth system sciences; 110. ISBN 978-3-540-74699-7.

[9] NIMA TECHNICAL REPORT TR8350.2. *Department of Defense World Geodetic System 1984*. Springfield (US-VA): National Geospatial-Intelligence Agency, 2000.

[10] KING, A. D. Inertial Navigation - Forty Years of Evolution. *GEC review*. London: General Electric Company, 1998, **13**(No. 3), 140 - 149.

[11] Schuler, M. (1923). Die Störung von Pendel und Kreiselapparaten durch die Beschleunigung des Fahrzeuges. *Physikalische Zeitschrift*. 24 (16). Retrieved 2008-12-02.

[12] FRADEN, Jacobs. *Handbook of modern sensors: physics, designs, and applications*. 3rd ed. New York: Springer-Verlag, c2004. ISBN 03-870-0750-4.

[13] TEWARI, Ashish. *Atmospheric and space flight dynamics: modeling and simulation with MATLAB and Simulink*. Boston (US-MA): Birkhäuser, c2007. Modeling and simulation in science, engineering & technology. ISBN 978-0-8176-4437-6.

[14] IFEACHOR, Emmanuel C a Barrie W JERVIS. *Digital signal processing: a practical approach*. 2nd ed. New York: Prentice Hall, 2002, xxiii, 933 p. ISBN 0201596199.

[15] TOMPKINS, Willis J. *Biomedical digital signal processing: C-language examples and laboratory experiments for the IBM PC*. Englewood Cliffs, N.J.: Prentice Hall, c1993. ISBN 0-13-067216-5.

[16] GREPL, Robert. *Kinematika a dynamika mechatronických systémů*. Vyd. 1. Brno: Akademické nakladatelství CERM, 2007, 158 s. ISBN 978-80-214-3530-8.

[17] PAUL, Richard P. *Robot manipulators: mathematics, programming, and control : the computer control of robot manipulators*. Cambridge (US-MA): MIT Press, 1981, 279 p. ISBN 026216082x.

[18] JÍRA, Josef, Michal MICKA a Pavel PUCHMAJER. *Kinematika a dynamika v dopravě: příklady*. 2., přeprac. vyd. Praha: České vysoké učení technické, 2010, 139 s. ISBN 978-80-01-04592-3.

[19] WOODMAN, O. J. *Technical Report: An introduction to inertial navigation*. Cambridge (UK): University of Cambridge, 2007. ISSN 1476-2986.

[20] GREWAL, Mohinder S., Lawrence R. WEILL a Angus P. ANDREWS. *Global positioning systems, inertial navigation, and integration*. 2nd ed. Hoboken, N.J.: Wiley-Interscience, c2007. ISBN 978-0-470-04190-1.

[21] GREWAL, Mohinder S. a Angus P. ANDREWS. *Kalman filtering: theory and practice using MATLAB*. 3rd ed. Hoboken, N.J.: Wiley, c2008. ISBN 978-0-470-17366-4.

[22] ECEF: LEICK, Alfred. *GPS satellite surveying*. 3rd ed. Hoboken, NJ: John Wiley, c2004. ISBN 0-471-05930-7.

[23] *Google Maps* [online]. Mountain View (US-CA): Google, 2017 [cit. 2017-08-15]. Dostupné z: http://maps.google.com/

[24] *OpenStreetMap* [online]. Sutton Coldfield (UK): OpenStreetMap Foundation, 2017 [cit. 2017-08-15]. Dostupné z: https://www.openstreetmap.org/

[25] *Mapy.cz* [online]. Praha: Seznam.cz, 2017 [cit. 2017-08-15]. Dostupné z: https://mapy.cz/

[26] Geoprohlížeč. *Český úřad zeměměřický a katastrální* [online]. Praha: Český úřad zeměměřický a katastrální, 2017 [cit. 2017-08-15]. Dostupné z: http://geoportal.cuzk.cz/geoprohlizec/

[27] BATTERSBY, Sarah E., Michael P. FINN, E. Lynn USERY a Kristina H. YAMAMOTO. Implications of Web Mercator and Its Use in Online Mapping. *Cartographica: The International Journal for Geographic Information and Geovisualization*. Canadian Cartographic Association, 2014, **49**(No. 2), 85 - 101. ISSN 0317-7173.

[28] *IOGP: About the EPSG Dataset* [online]. London: International Association of Oil & Gas Producers, 2017 [cit. 2017-08-15]. Dostupné z: http://www.epsg.org/

[29] USA. *NGA Advisory Notice on "Web Mercator": Application Risks to Operations and Adherence to DoD World Geodetic System 1984 (WGS 84)*. In: . Springfield (US-VA): National Geospatial-Intelligence Agency - Geomatics Office. Dostupné také z: http://earth-info.nga.mil/GandG/wgs84/web_mercator/

[30] IHDE, I., J. LUTHARDT, C. BOUCHER, P. DUNKLEY, E. GUBLER, B. FARRELL a J. A. TORRES. Coordinate Reference Systems used in Europe - Including Map Projections. *Institute for Environment and Sustainability: Map Projections for Europe*. European Communities, 2003, , 35 - 47. Dostupné z: http://www.ec-gis.org/

[31] Souřadnicové systémy. *Český úřad zeměměřický a katastrální* [online]. Praha: Český úřad zeměměřický a katastrální, 2016 [cit. 2017-08-15]. Dostupné z: http://geoportal.cuzk.cz/

[32] BOWDITCH, Nathaniel. *American Practical Navigator: An Epitome of Navigation*. Vol. 2 (2017 Edition). Springfield (US-VA): National Geospatial-Intelligence Agency, 2017.

[33] Length of a Degree of Latitude and Longitude. *Maritime Safety Information* [online]. Springfield (US-VA): National Geospatial-Intelligence Agency, 2017 [cit. 2017-08-15]. Dostupné z: https://msi.nga.mil/MSISiteContent/StaticFiles/Calculators/degree.html

[34] MUELLER, K. Tysen. *Low-cost, Drift-free DGPS Locomotive Navigation System: Final Report for High-Speed Rail IDEA Project 22*. Washington DC: IDEA Programs - Transportation Research Board, 2003.

[35] HENSEL, Stefan a Carsten HASBERG. Bayesian Techniques for Onboard Train Localization. *IEEE/SP 15th Workshop on Statistical Signal Processing*. Cardiff (UK): IEEE, 2009, , 361-364. ISBN 978-1-4244-2710-9.

[36] VIJAYAKUMAR, Jothi V N, Haibo ZHANG, Zhiyi HUANG a Adeel JAVED. A Particle Filter Based Train Localization Scheme Using Wireless Sensor Networks. *IEEE 11th International Conference on Dependable, Autonomic and Secure Computing*. Chengdu (Sichuan): IEEE, 2013, , 269-274. ISBN 978-1-4799-3381-5.

[37] *FT230X USB to basic UART IC Datasheet*. Version 1.3. Glasgow (UK): Future Technology Devices International Ltd., 2015.

[38] *STM32F205xx - STM32F207xx: ARM-based 32-bit MCU*. Rev. 12. STMicroelectronics, 2014.

[39] *National Instruments VISA* [online]. Austin (US-TX): National Instruments, 2014 [cit. 2017-08-15]. Dostupné z: https://www.ni.com/visa/

[40] *National Instruments LabVIEW* [online]. Austin (US-TX): National Instruments, 2017 [cit. 2017-08-15]. Dostupné z: http://www.ni.com/en-us/shop/labview.html

[41] *Canmore Electronics* [online]. Jhubei City (Taiwan): Canmore Electronics Co., Ltd., 2016 [cit. 2016-01-20]. Dostupné z: http://www.canmore.com.tw/

[42] *LSM330DLC: iNEMO inertial module: 3D accelerometer and 3D gyroscope*. Rev 2. STMicroelectronics, 2012.

[43] *NVS Technologies AG* [online]. Montlingen (Switzerland): NVS Navigation Technologies, 2012 [cit. 2017-08-15]. Dostupné z: http://www.nvs-gnss.com/

[44] *MCP73831/2: Miniature Single-Cell, Fully Integrated Li-Ion, Li-Polymer Charge Management Controllers*. Revision E. Chandler (US-AZ): Microchip Technology, 2008.

[45] *LIS344ALH: MEMS inertial sensor high performance 3-axis ±2/±6g ultracompact linear accelerometer*. Rev 3. STMicroelectronics, 2008.

[46] *LY330ALH: MEMS motion sensor: high performance ±300 dps analog yaw-rate gyroscope*. Rev 2. STMicroelectronics, 2010.

[47] *NV08C-CSM-BRD GNSS card: GPS/GLONASS/GALILEO/QZSS/SBAS RECEIVER*. Montlingen (Switzerland): NVS Technologies, 2013.

[48] *8117D GPS L1/GLONASS L1 Active: Magnetic Mount Antenna*. PCTEL. Dostupné z: http://www.antenna.com/

[49] *FatFs - Generic FAT Filesystem Module* [online]. Saitama (Japan): ChaN, 2017 [cit. 2017-08-15]. Dostupné z: http://elm-chan.org/fsw/ff/00index_e.html

[50] *SD Specifications Part E1: SDIO Simplified Specification*. Version 2.00. San Ramon (US-CA): SD Card Association, 2007.

[51] *The Kermit Project: The New Open-Source Kermit Project* [online]. New York: Kermitproject.org, 2017 [cit. 2017-08-15]. Dostupné z: http://www.kermitproject.org/

[52] JOHNSTON, Wesley M., J. R. Paul HANNA a Richard J. MILLAR. Advances in Dataflow Programming Languages. *ACM Computing Surveys*. New York: Association for Computing Machinery, 2004, **36**(No. 1), 1 - 34. ISSN 1557-7341.

Seznam autorových publikací

Statě ve sbornících

[A1] PUŠMAN, L. Aplikace principu fázového závěsu v řídicím algoritmu. In *Elektrotechnika a informatika 2011. Část 2., Elektronika*. Plzeň: Západočeská univerzita v Plzni, 2011. s. 89-92. ISBN: 978-80-261-0015-7

[A2] PUŠMAN, L. USB stack pro MCU Freescale. In *Elektrotechnika a informatika 2012. Část 2., Elektronika*. Plzeň: Západočeská univerzita v Plzni, 2012. s. 119-122. ISBN: 978-80-261-0119-2

[A3] PUŠMAN, L., KOSTURIK, K. Control Algorithm Based on Phase Locked Loop. In *21 st Telecommunications Forum (TELFOR) Proceedings of Papers*. Bělehrad: IEEE, 2013. s. 605-607. ISBN: 978-1-4799-1419-7

[A4] PUŠMAN, L. CNC odporová řezačka. In *Elektrotechnika a informatika 2013. Část 2., Elektronika*. Plzeň: Západočeská univerzita v Plzni, 2013. s. 81-84. ISBN: 978-80-261-0232-8

[A5] PUŠMAN, L., KOSTURIK, K. Integration of Digimatic Measuring Tool into LinuxCNC Controlled Milling Machine by Using MODBUS. In *2014 22nd Telecommunications Forum (TELFOR) Proceedings of Papers*. Bělehrad: IEEE, 2014. s. 691-693. ISBN: 978-1-4799-6190-0

[A6] PUŠMAN, L. Připojení měřicího přístroje s výstupem Digimatic k CNC frézce řízené softwarem LinuxCNC pomocí sběrnice MODBUS. In *Elektrotechnika a informatika 2014. Část 2, Elektronika*. Plzeň: Západočeská univerzita v Plzni, 2014. s. 57-60. ISBN: 978-80-261-0366-0

[A7] PUŠMAN, L., TURJANICA, P., KOSTURIK, K. Verification of Train Localization by Using Inertial Measurement System and Identification of Characteristic Segments. In *Proceedings of Papers : 2015 23rd Telecommunications Forum (TELFOR 2015)*. Piscataway: IEEE, 2015. s. 662-665. ISBN: 978-1-5090-0055-5

[A8] PUŠMAN, L. Přípravek pro inerciální měření. In *Elektrotechnika a informatika 2015. Elektrotechnika, elektronika, elektroenergetika*. Plzeň: Západočeská univerzita v Plzni, 2015. s. 191-194. ISBN: 978-80-261-0514-5

Výzkumné zprávy

[A9] CHLÁDEK, S., TURJANICA, P., PUŠMAN, L., ŠVARNÝ, J., KADLEC, T. *Design a implementace automatizovaného testovacího systému.* Výzkumná zpráva, č. 22190-054-2015. Plzeň : Západočeská univerzita v Plzni, 2015.

[A10] PUŠMAN, L. *Přehled nových lineárních LDO a impulzních stabilizátorů napětí pro automotive vydaných v roce 2014.* Výzkumná zpráva, č. 22110-VZ001-2015. Plzeň : Západočeská univerzita v Plzni, 2015.

[A11] KOLÁŘ, J., POLÁČEK, L., PUŠMAN, L., ŠESTÁK, M., TURJANICA, P. *Summary of modifications made on Bitcoin Automated Teller Machine in order to fulfill EMC requirements.* Výzkumná zpráva, č. 22190-003-2015. Plzeň : Západočeská univerzita v Plzni, 2015. 30 s.

[A12] KOLÁŘ, J., POLÁČEK, L., PUŠMAN, L., ŠESTÁK, M., TURJANICA, P. *The conformity assessment of Bitcoin Automated Teller Machine to standards CSN.* Výzkumná zpráva, č. 22190-002-2015. Plzeň : Západočeská univerzita v Plzni, 2015. 30 s.

[A13] TURJANICA, P., DANIHLÍK, P., CHLÁDEK, S., PUŠMAN, L. *Graphical user interface for automated test system.* Výzkumná zpráva, č. 22190-045-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A14] CHLÁDEK, S., TURJANICA, P., ŠVARNÝ, J., PUŠMAN, L., KADLEC, T., MÁCHA, V., DRNEK, J. *Implementace systému, pilotní provoz, ověření výsledků.* Výzkumná zpráva, č. 22190-049-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

Funkční vzorky a prototypy

[A15] PUŠMAN, L., KOSTURIK, K. *Řídicí jednotka motorů pro mobilní platformu.* Funkční vzorek. Číslo technické dokumentace: 22110-FV028-2011. Plzeň : Západočeská univerzita v Plzni, 2011.

[A16] PUŠMAN, L., KOSTURIK, K. *Modul převodníku USB/CAN s MCU Freescale a implementací USB stacku.* Funkční vzorek. Číslo technické dokumentace: 22110-FV050-2012. Plzeň : Západočeská univerzita v Plzni, 2012.

[A17] PUŠMAN, L., KOSTURIK, K. *CNC odporová řezačka.* Funkční vzorek. Číslo technické dokumentace: 22110-FV001-2013. Plzeň : Západočeská univerzita v Plzni, 2013.

[A18] PUŠMAN, L. *Připojení digitálního úchylkoměru Mitutoyo s výstupem Digimatic k řídicímu softwaru LinuxCNC pomocí sběrnice MODBUS.* Funkční vzorek. Číslo

technické dokumentace: 22110-FV012-2014. Plzeň : Západočeská univerzita v Plzni, 2014.

[A19] TURJANICA, P., ŠVARNÝ, J., PUŠMAN, L., CHLÁDEK, S. *Jednotka pro vkládání chybových stavů a sběr dat.* Funkční vzorek. Číslo technické dokumentace: 22190-FV030-2015. Plzeň : Západočeská univerzita v Plzni, 2015.

[A20] POLÁČEK, L., JÁRA, M., PUŠMAN, L., BURIAN, P. *Prototype of the core modules of the modular control HW and ECU.* Funkční vzorek. Číslo technické dokumentace: 22190-FV021-2015. Plzeň : Západočeská univerzita v Plzni, 2015.

[A21] PUŠMAN, L. *Přípravek pro inerciální měření založený na obvodech LSM330DLC.* Funkční vzorek. Číslo technické dokumentace: 22110-FV001-2015. Plzeň : Západočeská univerzita v Plzni, 2015.

[A22] PUŠMAN, L., POLÁČEK, L., JÁRA, M., BURIAN, P. *Přípravek pro ověření vlastností izolačního převodníku ACPL-C797.* Funkční vzorek. Číslo technické dokumentace: 22190-FV004-2015. Plzeň : Západočeská univerzita v Plzni, 2015.

[A23] PUŠMAN, L., TURJANICA, P., CHLÁDEK, S. *Karta Signal Conditioning Unit a příslušenství.* Funkční vzorek. Číslo technické dokumentace: 22190-FV001-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A24] TURJANICA, P., CHLÁDEK, S., PUŠMAN, L., ŠVARNÝ, J. *Pilot automated test system prototype.* Funkční vzorek. Číslo technické dokumentace: 22190-FV056-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A25] POLÁČEK, L., MOLNÁR, J., VOŠMIK, D., PUŠMAN, L. *Regulovaný zdroj stejnosměrného napětí FU5A.* Funkční vzorek. Číslo technické dokumentace: 22190-FV041-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A26] PUŠMAN, L., TURJANICA, P. *Řídicí elektronika pro automatizovaný terč Mitteo.* Funkční vzorek. Číslo technické dokumentace: 22190-FV018-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A27] POLÁČEK, L., MOLNÁR, J., KOŠAN, T., PUŠMAN, L. *Řídící jednotka pro měření parametrů sítě CI5.* Funkční vzorek. Číslo technické dokumentace: 22190-FV043-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A28] PUŠMAN, L., FREISLEBEN, J., ČENGERY, J., KAŠPAR, P. *Senzor teploty, vlhkosti a tlaku pro síť LoRaWAN.* Funkční vzorek. Číslo technické dokumentace: 22130-FV009-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A29] POLÁČEK, L., MOLNÁR, J., KOŠAN, T., PUŠMAN, L., KOMRSKA, T. *Výkonová jednotka pro měření parametrů sítě CI5.* Funkční vzorek. Číslo technické dokumentace: 22190-FV044-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A30] TURJANICA, P., CHLÁDEK, S., PUŠMAN, L., ŠVARNÝ, J., KUČERA, V., MÁCHA, V. *Tester pro konečnou validaci výrobků ve výrobě.* Prototyp. Číslo technické dokumentace: 22190-PR005-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

Software

[A31] TURJANICA, P., DANIHLÍK, P., PUŠMAN, L., CHLÁDEK, S. *Graphical user interface for automated test system.* Software. Číslo technické dokumentace: 22190-SW007-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

[A32] CHLÁDEK, S., PUŠMAN, L. *Knihovny ovladačů speciálních periferií.* Software. Číslo technické dokumentace: 22190-SW008-2016. Plzeň : Západočeská univerzita v Plzni, 2016.

Ostatní a mimo RIV

[A33] TURJANICA, P., CHLÁDEK, S., PUŠMAN, L., NOVOTNÝ, J., VESELÝ, L., ŘEDINA, J. *CIDAM - WP04 : souhrnná zpráva 2016.* Č. 22190-050-2016 . Plzeň : Západočeská univerzita v Plzni, 2016.

[A34] Review článku T. Gahlot, A. Kamat and P. Swain, Automating toolpath generation for 3-Axis CNC. *2016 IEEE International Conference on Industrial Technology (ICIT)*, Taipei, 2016, pp. 836-841. doi: 10.1109/ICIT.2016.7474860

Připravované publikace

[A35] KAVALÍR, T., KINDL, V., PUŠMAN, L., TURJANICA, P. Patent v oblasti bezkontaktního přenosu elektrické energie na rotující součást, přihláška podána 9/2017

[A36] KAVALÍR, T., KINDL, V., PUŠMAN, L., TURJANICA, P. Užitný vzor v oblasti bezkontaktního přenosu elektrické energie na rotující součást, přihláška podána 9/2017

[A37] BURIAN, P., GEORGIEV, V., BERGMANN, L., V., ZICH, J., PUŠMAN, L., BROULÍM, P., POSPÍŠIL, S. *General-Purpose Solution for Timepix3 - Katherine Readout*, článek a poster přijaté na konferenci *Topical Workshop on Electronics for Particle Physics (TWEPP 2017)*, 9/2017, Santa Cruz (US-CA)