



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Platforma průmyslové spolupráce

CZ.1.07/2.4.00/17.0041

Název

Krokoměr pro Android

Popis a využití

- use case využití mobilních senzorů
- výuka: programování pro Android

Jazyk textu

- slovenský

Autor (autoři)

- Martin Lisý

Oficiální stránka projektu:

- <http://lasaris.fi.muni.cz/pps>

Dostupnost výukových materiálů a nástrojů online:

- <http://lasaris.fi.muni.cz/pps/study-materials-and-tools>

Obsah

1	Úvod	1
2	Inerciálna navigácia	3
2.1.	História použitia.....	3
2.2.	Metódy určenia polohy	4
2.2.1.	Wi-Fi lokalizácia	4
2.2.2.	Krokomer – lokalizácia senzormi.....	5
3	Analýza	6
3.1.	Jama.....	6
3.2.	Chyby	6
3.2.1.	Ludská a náhodná chyba.....	6
3.2.2.	Šum	7
3.2.3.	Odchýlka.....	7
3.2.4.	Posunutie nulového bodu	8
3.2.5.	Časové oneskorenie.....	8
3.2.6.	Chyba integrálu	8
4	Možnosti riešenia chýb	9
4.1.	Fúzia	9
4.2.	Kalibrácia	10
4.3.	Filtrácia dát	10
4.3.1.	Kálmánov filter	10
4.3.2.	Low-Pass filter.....	12
5	Implementácia	14
5.1.	Logické rozdelenie tried	14
5.2.	Detekcia krokov	16
5.3.	Určenie prejdenej vzdialenosti	17
5.3.1.	Výpočet vzdialenosti integrálom.....	17
5.3.2.	Fixná dĺžka kroku	18
5.4.	Grafické prostredie	19
6	Testovanie	22
7	Záver	25

1 Úvod

Určovanie polohy prostredníctvom GPS je vďaka satelitom obiehajúcim okolo Zeme jednoduchá záležitosť. Svoje uplatnenie našiel na komerčnom trhu hlavne v oblasti dopravy, no keďže sa GPS senzor nachádza už aj vo väčšine mobilných zariadení, napomáha aj pri hľadaní stratených, či ukradnutých prístrojov. Presnosť na jednotky metrov, ktorú globálny lokalizačný systém poskytuje, je vhodná hlavne na lokalizáciu v širšom priestranstve. No obiehajúce satelity nie je možné využiť na určenie polohy v uzatvorenom priestore. Komunikácia medzi zariadením a satelitom prebieha rýchlosťou svetla. Steny budov túto rýchlosť značne znižujú. Následkom toho dochádza k výraznej strate presnosti určovanej polohy. K určeniu polohy sú tiež nevyhnutné aspoň 4 rôzne satelitné signály, ktorým ale bráni prítomnosť okolitých stien. Preto treba využiť alternatívne spôsoby.

Mobilné zariadenia v súčasnosti okrem lokalizačného senzora obsahujú aj celý rad iných sensorov. No nebolo tomu vždy tak. S vývojom operačného systému Android sa do zariadení postupne pridávali rôzne senzory mapujúce fyzikálne vlastnosti okolia. Od verzie Android 1.5 bol pridaný základný balík sensorov, ktorý bol neskôr vo verzii Android 2.3 rozšírený o nové senzory tvoriace súčasný štandard. Tento štandard zahŕňa akcelerometer, magnetometer, gyroskop, svetelný senzor, tlakový senzor, senzor relatívnej vlhkosti a senzor blízkosti určujúci bez dotyku prítomnosť okolitých objektov [1]. Vďaka tomuto štandardu by malo byť možné určovať polohu vo vnútri budov bez využívania lokalizačných satelitov. Okrem využitia sensorov na vnútornú navigáciu existujú aj lokalizačné spôsoby založené na bezdrôtových sieťach. Využívanými sú hlavne Wi-Fi a Bluetooth.

Cieľom tejto práce je vytvoriť aplikáciu pre mobilné zariadenia pracujúce na platforme Android. Aplikácia má slúžiť ako krokomer a stať sa súčasťou vnútorného lokalizačného systému. Samotná aplikácia je schopná pracovať aj samostatne bez ďalších nastavieb a môže byť bežne používaná ako krokomer napríklad pre bežcov. Spomenuté senzory akcelerometer, magnetometer a gyroskop možno v konečnom dôsledku použiť na zmeranie počtu krokov, či určenie

vzdialenosti od východiskového bodu. Jednotlivé senzory však musia byť pred použitím kalibrované. Po kalibrácii prichádza na rad fúzia, teda funkčné zlúčenie hodnôt prúdiacich z jednotlivých sensorov do použiteľnej podoby. Sensory sú schopné kalibrovať sa fúziou navzájom. Uvažujme, že prítomnosť kovového predmetu môže magnetometer vyhodnotiť ako zmenu smeru, aj keď je zariadenie v pokoji. Gyroskop však v rovnakom čase zmenu smeru nezaznamená. Sensory sa tak vzájomne dopĺňajú pri určení orientácie. Pri získavaní údajov o zrýchlení z akcelerometra, o orientácii z magnetometra a o natočení telefónu či tabletu z gyroskopu sa okrem samotných dát hromadia aj rôzne šумы a akumulované chyby. Okrem prítomnej odchýlky merania sensorov dochádza aj k zaznamenávaniu jemných pohybov spôsobených trasením rúk. Nezanedbateľným prvkom je aj všadeprítomná gravitačná sila, ktorej zložka sa pričíta k meraniu zrýchlenia. Následne je potrebné gravitačný vektor odčítať. Tieto nedostatky spôsobujú výrazné nepresnosti pri samotnom meraní počtu krokov a vzdialenosti.

Vzhľadom na spomenuté skutočnosti je nutné použiť Bayesiánske filtre. Tieto filtre redukujú šum a chyby. Konkrétne Kálmánov filter berie na vstupe dáta zo sensorov a na výstup poskytuje dáta upravené príslušným algoritmom. Udržiava mieru pravdepodobnosti obdržaných dát v čase. Tým eliminuje náhle výkyvy signálu od normálneho priebehu.

Po odstránení nedokonalostí dát (aj keď stopercentné odstránenie nie je nikdy možné) sú následne údaje spracované na základe príslušných podmienok. Podľa nich sa určí, či pri výkyve zariadenia nastal krok alebo nie. Dochádza hlavne k porovnaniu aktuálnej hodnoty zrýchlenia a predom stanovenej minimálnej hodnoty zrýchlenia pre krok. Porovnáva sa tiež časový rozdiel medzi dvoma krokmi. Okrem spomenutých podmienok do procesu detekcie krokov vstupujú aj ďalšie dôležité ukazovatele ako nárast či pokles zrýchlenia.

Predmetom skúmania tejto práce je tiež meranie prejdenej vzdialenosti rôznymi spôsobmi. Tými spôsobmi môžu byť dvojný integrál zrýchlenia cez čas alebo súčin fixnej dĺžky kroku a počtu prejdenej krokov. Práca sa zaoberá hlavne metódou integrovania.

2 Inerciálna navigácia

Inerciálny navigačný systém¹ je lokalizačný systém určujúci polohu prenosného zariadenia. Využíva vstavané pohybové a rotačné senzory na určenie orientácie, zrýchlenia a prítomnosti magnetického poľa. Vychádza z predpokladu, že iniciálna poloha, rýchlosť a orientácia sú pri začiatku navigácie známe. INS využívajú lode, ponorky, lietadlá, rôzne navádzané strely. S rozvojom technológií sa však INS stáva aj súčasťou mobilných prístrojov, kde môže fungovať jednoducho ako navigačný systém pri hľadaní konkrétneho miesta v budove.

2.1 História použitia

Z historického hľadiska bol inerciálny navigačný systém pôvodne vymyslený pre rakety. Počas druhej svetovej vojny bolo zámerom vytvoriť raketu dlhého doletu, ktorá by bola schopná sa sama navigovať. Nemecký Wernher von Braun zostrojil raketu V2², v ktorej skombinoval funkčnosť dvoch voľných gyroskopov (horizontálneho a vertikálneho) a PIGA akcelerometra³. Gyroskopy plnili funkciu postrannej stabilizácie a PIGA akcelerometer po dosiahnutí stanovenej rýchlosti odpojil motor [2]. Komerčné využitie našiel INS v oblasti letectva. Pred zavedením komplexných riadiacich systémov v leteckej doprave boli lietadlá vybavené systémom „Delco Carousel“. Delco Carousel je automatický navigačný systém založený na INS, ktorý dovoľoval pilotom zadávať body trasy. Podľa zadaných bodov systém spočítal pozíciu a rýchlosť lietadla a následne ho dokázal automaticky navigovať. Neskôr tento systém odkúpila spoločnosť Boeing, ktorá vo svojich prvých modeloch lietadiel využívala práve navigáciu inšpirovanú Delco Carouselom. S rozvojom techniky našiel gyroskop a akcelerometer svoje uplatnenie aj v mobilných zariadeniach. Okrem primárneho účelu

¹ INS - Inertial navigation system

² nemecky Vergeltungswaffe 2, Odvetná zbraň 2

³ akcelerometer zaznamenávajúci zrýchlenie a simultánne integrujúci toto zrýchlenie cez čas za účelom získania hodnoty rýchlosti

sledovať orientáciu zariadenia a otáčať podľa orientácie zobrazované dáta na displeji môžu byť tieto senzory využité tiež pri implementácii funkčného INS.

2.2 Metódy určenia polohy

Jedným z mnohých spôsobov určenia polohy v budove je určovanie pozície prostredníctvom výpočtu zmeny vzdialenosti od východiskového stanoviska. Potrebné vstupné údaje poskytujú vstavané senzory zariadenia a externý zdroj určujúci iniciálnu polohu. Druhým spôsobom je určovanie pozície žiadaním dostupných Wi-Fi routerov o svoju aktuálnu polohu.

2.2.1 Wi-Fi lokalizácia

Začiatkom 21. storočia neboli obvyklým vybavením budov Wi-Fi routere. No dnes by sme len ťažko našli budovu bez prístupových bodov. Odhliadnuc od ich primárneho účelu pristupovať na internet, možno zužitkovať vysielaný signál aj iným spôsobom. Zariadenia vybavené prijímačom signálu môžu byť lokalizované Wi-Fi routerom. Samotný proces lokalizácie je však veľmi náročný, nakoľko signálu prekrážajú podlahy, steny budov či ľudia pohybujúci sa v okolí.

Určenie polohy prebieha dvojako. Buď sa na výpočet pozície použijú parametre ako TOA⁴, TDOA⁵ a TDOA⁶. Táto metóda však funguje len pri bezbarierovej komunikácii a presnej synchronizácii medzi vysielateľom a prijímačom. Alebo môže byť zariadenie lokalizované využitím hodnoty sily prichádzajúceho signálu z vysielateľa. Metóda využíva úroveň sily signálu snímané mobilnou stanicou [3]. Lokalizačný systém založený na signálových odtlačkoch musí najskôr zozbierať presné údaje o hodnotách síl signálu z okolia routerov, ktoré pokrývajú danú oblasť budovy. Zbieranie údajov je

⁴ The time of arrival - čas príchodu signálu

⁵ The time difference of arrival - časový rozdiel príchodu signálu

⁶ The direction of arrival - smer príchodu signálu

však časovo náročné a vplyvom meniacich sa podmienok oblasti sú uložené dáta znehodnocované. Počet ľudí v oblasti, premiestnenie alebo nákup nového nábytku do budovy značne ovplyvňuje hodnoty sily signálu. Vykonané zmeny v dispozícií vyžadujú nové merania na udržanie aktuálnosti databázy hodnôt síl signálu. Nedá sa tiež opomenúť potreba prítomnosti väčšieho počtu prístupových bodov v budove, čo môže byť problémom napríklad v prípade historických budov alebo nemocníc [4]. Samotná Wi-Fi lokalizácia je vzhľadom na spomenuté skutočnosti vhodná skôr ako doplnková funkcia pri lokalizácií zariadenia v budove. Prístroju oznámi jeho približnú polohu, ktorú ďalej spracúva INS v spolupráci s vnútornými senzormi mobilnej stanice.

2.2.2 Krokomer – lokalizácia senzormi

Inerciálny navigačný systém spája Wi-Fi lokalizáciu, ktorá je podporným systémom detekujúcim iniciálnu a prípadne aj priebežnú pozíciu, s krokomerom slúžiacim na určovanie počtu krokov a smeru pohybu. Krokomer principiálne funguje vďaka výpočtovej jednotke IMU⁷. Tá sa skladá z trojosového akcelerometra. Akcelerometer zaznamenáva zrýchlenie pozdĺž každej z troch priestorových osí. Okrem akcelerometra je na ďalšie výpočty nutný trojosový gyroskop sledujúci orientáciu zariadenia, takisto každý pozdĺž inej priestorovej osy. Smer a veľkosť magnetickej indukcie meria trojosový magnetometer. Spracovaním týchto deviatich signálov dostávame možnosť sledovať polohu zariadenia. K sledovaniu polohy ešte potrebuje dodatočné informácie, konkrétne súradnice iniciálnej pozície, od ktorej sa bude zaznamenávať zmena polohy. INS nedokáže určiť zemepisnú šírku a dĺžku umiestnenia prístroja. Takúto informáciu poskytuje zvyčajne externý zdroj, napríklad práve dostupný Wi-Fi router. Dôležitými sú tiež počiatočná orientácia prístroja a jeho začiatočná rýchlosť [5].

Výhodou INS je fakt, že na určenie polohy zariadenie nepotrebuje žiadne externé zdroje informácií o zrýchlení alebo orientácií. Všetky potrebné údaje si dokáže po inicializácii pozície, orientácie a rýchlosti zabezpečiť samo.

⁷ Inertial measurement unit – vnútorná výpočtová jednotka

3 Analýza

Analýza sa zaoberá najčastejšími chybami, ktoré vznikajú počas čítania dát zo senzorov a ich prípadnou elimináciou. Eliminácia týchto chýb môže výrazne ovplyvniť beh programu. Aplikácia využíva Jama⁸ triedu, ktorá zabezpečuje potrebné zložité výpočty.

3.1 Jama

Počas behu aplikácie prebieha mnoho zložitých výpočtov, ktoré filtrujú dáta. Trieda Matrix poskytuje numerické operácie lineárnej algebry realizujúce tieto výpočty. Konštruktory vytvárajú matice z dvojrozmerných polí. Okrem jednoduchých operácií sčítania a násobenia matic trieda obsahuje aj maticové normy.

3.2 Chyby

Pred samotnou implementáciou jednotlivých častí aplikácie je nutné spomenúť, akým druhom chýb bude potrebné čeliť. Pozorovanie a následná náprava vyskytovaných chýb výrazne zvyšuje efektívnosť určovania počtu krokov a prejdenej vzdialenosti. Podľa [1] sú najdôležitejšími chybami, ktoré treba pred samotnou implementáciou pochopiť, nasledujúce:

3.2.1 Ľudská, systematická a náhodná chyba

Ľudská chyba je chyba spôsobená človekom. Neželané vychýlenie od reality privodia buď pád mobilného zariadenia, alebo

⁸ Java Matrix Class

prudké pohyby rukou užívateľa. V takýchto situáciách môže dôjsť k započítaniu falošného kroku, ktorý v skutočnosti nebol vykonaný.

Systematická chyba je chyba ovplyvňujúca presnosť výpočtov. Jedná sa o periodické vychýlenie o konštantnú odchýlku od skutočnej hodnoty. Takáto chyba sa môže objaviť napríklad pre meraní magnetometrom v blízkosti kovových predmetov alebo magnetov. Chyba môže byť v niektorých prípadoch eliminovaná kalibráciou či fúziou senzorov.

Náhodná chyba, ako napríklad šum, sa odzrkadľuje na presnosti merania. Keďže výskyt chyby je náhodný, nie je možné ju predvídať alebo algoritmicky eliminovať.

3.2.2 Šum

Šum je nechcená modifikácia signálu, ktorá vzniká počas vzorkovania. Tvoria ho rôzne výkyvy počítanej hodnoty. Niekedy sa šumom označujú aj náhodné nepredvídateľné signály, ktoré neobsahujú žiadnu užitočnú informáciu. Ak šum nie je náhodný, je možné poškodený signál opraviť použitím špeciálnych filtrov. Šum signálu môže byť špecifikovaný jeho štatistickými vlastnosťami, ktoré sú známejšie pod pojmom farba šumu. Príkladom je biely šum, čo je náhodný signál s konštantne silnou spektrálnou hustotou.

3.2.3 Odchýlka

Odchýlkou sa nazýva pomalé, dlhodobé vychýľovanie dát od ich skutočnej hodnoty. Odchýlka sa môže objaviť počas čítania dát zo senzorov kvôli degradácii signálu vplyvom času. Môže sa tiež objaviť pri integrovaní hodnôt. V prípade výskytu odchýlky sa v každej iterácii integrovania pridáva malá kompenzačná hodnota. Tá spôsobí vo výslednom čítaní dát zo senzorov vychýlenie od skutočného merania, čím odchýlku vykompenzuje.

3.2.4 Posunutie nulového bodu⁹

Ak by merané vlastnosti mali byť nulové, no výstupné dáta prúdiace zo senzorov nulové nie sú, jedná sa o posunutie nulového bodu. Príkladom je gravitačné zrýchlenie. Ak je mobilné zariadenie v pokoji, bez akéhokoľvek pohybu, akcelerometer by mal na výstupe poskytovať hodnoty (0; 0; 0). Vplyvom gravitačného zrýchlenia ale nastáva vychýlenie na jednej z meraných osí, a preto namiesto toho zo senzorov prúdia hodnoty (0; 0; $-9,80665 \text{ ms}^{-2}$).

3.2.5 Časové oneskorenie

Keďže Android nie je RTOS¹⁰, pri čítaní dát zo senzorov dochádza k malému oneskoreniu, čo vedie k nepresným časovým odtlačkom. Keď je zariadenie zaneprázdnené, môže dôjsť aj k vynechaniu niektorých dát. Tieto chyby nemožno nijako ovplyvniť.

3.2.6 Chyba integrálu

Gyroskop poskytuje dáta o uhlovej rýchlosti v radiánoch za sekundu. V aplikáciách je ale užitočnejším údajom uhol v radiánoch, o ktorý sa zariadenie vychýlilo z pôvodného stavu. Na získanie uhlu treba integrovať uhlovú rýchlosť cez čas. Lenže jednoduchá integrácia neposkytuje dostatočne presné výsledky. Odchýlky a posunutia nulového bodu tieto čísla výrazne ovplyvňujú. Postupom času dochádza k narastaniu rozdielu oproti realite a aktuálne hodnoty nezodpovedajú skutočnému natočeniu zariadenia. Ideálnym riešením tohto problému je fúzia senzorov. Spomenuté chyby sa tak kompenzujú využívaním dát z iných dostupných senzorov, hlavne z akcelerometra.

⁹ Z ang. Zero Offset

¹⁰ Real Time Operating System – Operačný systém spracúvajúci požiadavky v reálnom čase

4 Možnosti riešenia chýb

Všetky problémy spomínané v predchádzajúcej kapitole výrazným spôsobom ovplyvňujú detekciu krokov a aj meranie prejdenej vzdialenosti. Práve preto je nutné sa týmito problémami zaoberať a riešiť ich. Väčšinu problémov je možné eliminovať, no stále zostáva súčasťou signálu napríklad náhodná chyba či trasenie rúk užívateľa. Na elimináciu slúžia fúzia, kalibrácia a filtrácia dát.

4.1 Fúzia

Proces kombinovania dát prúdiacich z viacerých senzorov za účelom dosiahnuť lepšie výsledky sa označuje ako fúzia senzorov. Spojenie viacerých senzorov zabezpečí využitie silných stránok jednotlivých senzorov a minimalizuje ich slabosti. Napríklad akcelerometer rýchlo reaguje na zmeny, ale dáta, ktoré zaznamenáva, obsahujú veľa šumov. Vyhladzovanie týchto šumov spôsobuje neželané oneskorené reakcie zariadenia. Integrovanie prichádzajúcich hodnôt z gyroskopu cez čas za účelom získania rýchlosti alebo prejdenej vzdialenosti poskytuje výpočty presnejšie, s nižšou hladinou šumov. Tieto údaje sú však častokrát samostatne nepoužiteľné, lebo integrované hodnoty dát z gyroskopu sa rýchlo začínajú výrazne odlišovať od reálnej orientácie prístroja. Vzhľadom na spomenuté fakty by pri fúzii senzorov mali byť využívané hlavne integrované dáta z gyroskopu. Rýchlo narastajúca odchýlka od reálnej orientácie musí byť korigovaná pravidelným porovnávaním gyroskopických hodnôt s hodnotami z akcelerometra, ktoré takejto odchýlke nepodliehajú [1]. Ďalším príkladom funkčného spojenia dvoch senzorov je spojenie magnetometra a akcelerometra. V tomto prípade akcelerometer dokáže určiť, že sa zariadenie hýbe, ale nedokáže určiť, na ktorú svetovú stranu smeruje. Informáciu o orientácii na svetové strany však dokáže dodať magnetometer, ktorý na to primárne slúži.

4.2 Kalibrácia

Po prvom spustení aplikácie sú senzory zariadenia kalibrované. Kalibrácia eliminuje výskyt systematickej chyby (viď Ľudská, systematická a náhodná chyba, kapitola 3.2.1), ktorá môže byť prítomná pri čítaní dát zo sensorov. Zariadenie je uvedené do pokoja vzhľadom na všetky osi a zo sensorov sa čítajú informácie. Namerané hodnoty udávajú veľkosť systematickej chyby. Tieto čísla sú následne za behu programu pri každom čítaní dát zo sensorov odrátané, čím sa systematická chyba odstráni a zvýši sa presnosť meraní.

4.3 Filtrácia dát

Ďalšou z techník využívanou pri oprave a vyhladení nedokonalostí signálov prúdiacich zo sensorov je filtrácia dát. Filtre sú algoritmy, ktorými sa z nespracovaných hodnôt plných rôznych chýb dajú získať žiadané elementy originálnych dát. Aplikácia využíva nasledujúce filtre:

4.3.1 Kálmánov filter

Kálmánov filter je digitálny filter používaný na filtrovanie chýb a šumov. Pracuje rekurzívne na dátovom toku dynamického systému, čím sa získa optimálny odhad aktuálneho stavu systému. Ten má početné uplatnenie napríklad v leteckej navigácii alebo navigačných kontrolných systémoch. Svoje uplatnenie nachádza aj všade tam, kde je nutné z prúdiaceho toku dát odstrániť neželané rušivé elementy. Ako tvrdí Singhal v [7], použitiu Kálmánovho filtra predchádza znalosť vlastností zdrojového signálu. Dôležitými ukazovateľmi sú ako sa skutočná meraná hodnota správa, alebo čo spôsobuje zmenu hodnôt. Algoritmus filtra je založený na dvojfázovom spracovaní prúdiacich dát. V prvej fáze algoritmus odhaduje hodnotu stavových veličín, ktoré zahŕňajú aj náhodný šum.

Takýto odhad obsahuje iba dáta zaznamenané pred spracovaním aktuálnej vzorky. Odhad je vypočítaný vždy pred spracovaním každej novej vzorky, aby mohol byť správne určený aktuálny stav systému. V druhej fáze algoritmus odhadu pracuje s váženým priemerom predpovedanej a aktuálnej hodnoty. Čím je aktuálna hodnota bližšie k predpovedanej, tým dostáva vo výsledku vyššiu váhu.

Realizácia Kálmánovho filtra vychádza z [8] a využíva pri výpočtoch nasledujúce vzťahy:

$$\mathbf{x}_k = \mathbf{A} \times \mathbf{x}_{k-1} + \mathbf{B} \times \mathbf{u}_k + \mathbf{w}_k \quad (1)$$

$$\mathbf{z}_k = \mathbf{H} \times \mathbf{x}_k + \mathbf{v}_k \quad (2)$$

kde:

\mathbf{x}_k (\mathbf{x}_{k-1})	aktuálny stav systému v čase k (k-1)
\mathbf{z}_k	výpočet stavu systému v čase k
\mathbf{u}_k	externe kontrolovaný vstup v čase k
\mathbf{v}_k a \mathbf{w}_k	chyby počas výpočtov a distribúcií signálu
A	matica váhy spájajúca stav systému v čase k-1 so stavom v čase k
B	matica váhy spájajúca kontrolovaný vstup \mathbf{u}_k s aktuálnym stavom \mathbf{x}_k
H	matica váhy spájajúca aktuálny stav \mathbf{x}_k s výpočtom stavu \mathbf{z}_k

Matice A, B a H sú automaticky inicializované po alokácii Kálmánovho filtra konštruktorom.

Algoritmus pracuje rekurzívne. Rekurzia Kálmánovho filtra spočíva v tom, že na výpočet aktuálneho stavu je potrebný len

odhadovaný stav z predchádzajúceho časového okamihu a aktuálny výpočet. Informácie z predchádzajúcich stavov sú zahrnuté v každom z odhadov. Filter nevyžaduje zaznamenávanie žiadnych ďalších údajov ani informácie o predchádzajúcich meraniach. Výpočty prebiehajú v cykloch používajúc jednoduché rovnice. Takýto nenáročný algoritmus zbytočne nezaťažuje systém, a teda je relatívne efektívny.

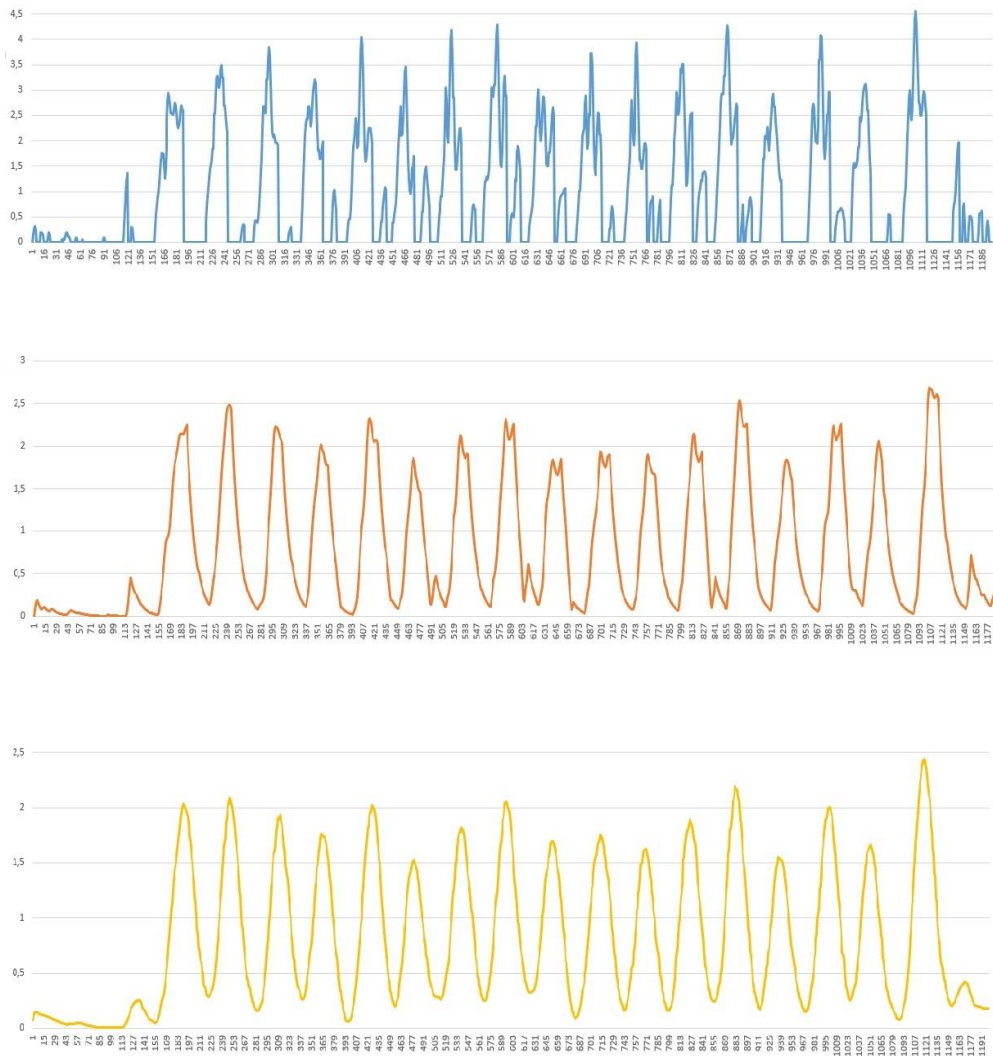
4.3.2 Low-Pass filter

Low-Pass¹¹ filter oddeľuje signály s nižšou frekvenciou od signálov s frekvenciou vyššou, akou je predom stanovená hodnota. Ideálny filter by mal eliminovať všetky signály o frekvencii vyššej ako hraničná hodnota a ponechať všetky signály s frekvenciou nižšou. Ideálny filter môže byť realizovaný matematicky, a to vynásobením hodnoty signálu obdĺžnikovou funkciou¹² vo frekvenčnej doméne. Ideálne podmienky filtru je však nemožné dosiahnuť bez signálov nekonečného rozsahu v čase. Skutočné filtre pre reálne aplikácie aproximujú ideálny filter vytvorením okna, mimo ktorého sú nekonečné signály nahradené za nulovú hodnotu.

Na nasledujúcom obrázku (obr. 4.3.2.1) možno pozorovať vplyv filtrov na dáta prúdiace zo sensorov. Modrý graf reprezentuje hodnoty neupravené filtrami, hnedý graf je výstupom Kálmánovho filtra. Graf výstupných dát je vyhladený, šumy a chyby sú eliminované. Následne je aplikovaný Low-Pass filter, ktorý dáta ešte finálne upraví do výslednej podoby (žltý graf).

¹¹ Vysvetlenie názvu Low-Pass filter podľa [1] - **passes low frequency signal** (ponecháva nízkofrekvenčný signál)

¹² Podľa [6] definícia obdĺžnikovej funkcie: $rect(t) = \Pi(t) = \begin{cases} 0 & ak |t| > \frac{1}{2} \\ \frac{1}{2} & ak |t| = \frac{1}{2} \\ 1 & ak |t| < \frac{1}{2} \end{cases}$



Obr. 4.3.2.1 Reprezentácia dát v ms^{-2} vstupných (modrý graf), filtrovaných Kálmánovým filtrom (hnedý graf) a výsledný graf po aplikácii Kálmánovho a Low-Pass filtra (žltý graf)

5 Implementácia

Aplikácia je implementovaná tak, aby bola kompatibilná s väčšinou aktuálne dostupných zariadení. Cieľovou skupinou sú zariadenia pracujúce s programovým rozhraním API level 18 (Android verzia 4.3). Samozrejماً je aj spätná kompatibilita so staršími verziami tohto operačného systému. Požiadavkou je však minimálne programové rozhranie API level 9 (Android verzia 2.3).

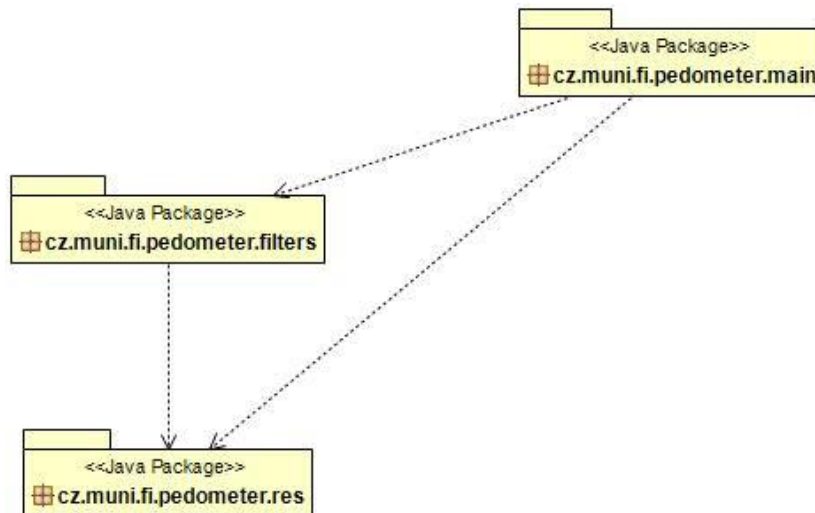
5.1 Logické rozdelenie tried

Vzhľadom na funkčnosť jednotlivých tried aplikácie sú triedy podľa významovo podobnej funkcionality rozdelené do balíkov. Balíky prinášajú do programu väčšiu prehľadnosť a organizovanosť. Vzťahy medzi balíkmi možno vidieť na obr. 5.1.1. Nasleduje zoznam balíkov aplikácie spolu s ich funkčným popisom:

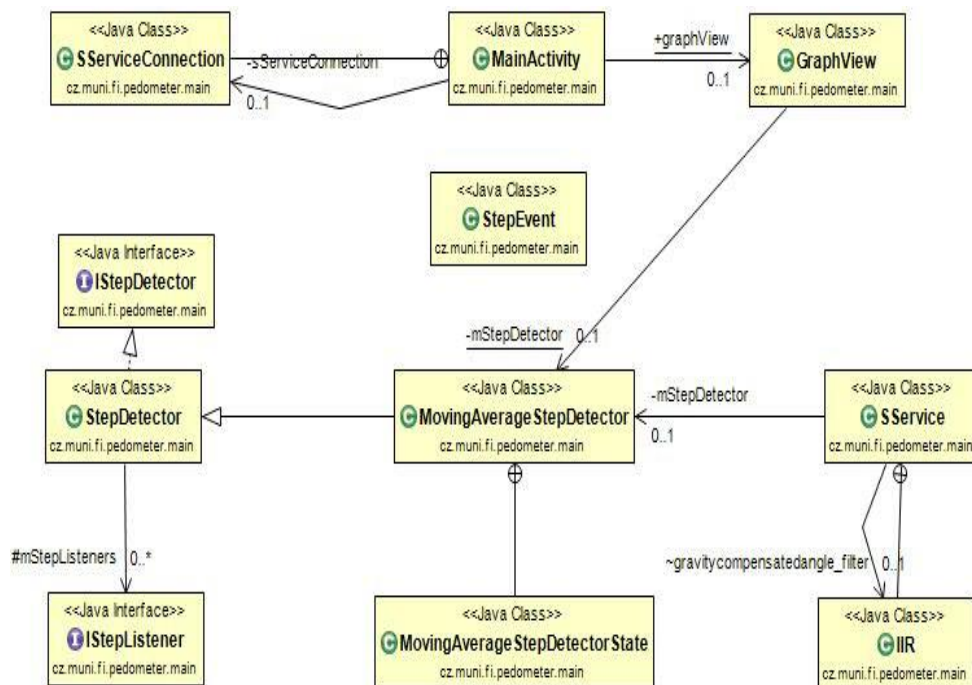
cz.muni.fi.pedometer.main – balík obsahuje hlavné triedy programu. Trieda *MainActivity* je vyhodnotená po spustení aplikácie. Okrem tejto triedy balík obsahuje aj dôležité triedy vykonávajúce funkcionality programu. Trieda *SService* je komponent aplikácie spracúvajúci dáta prúdiace z jednotlivých senzorov. Grafické zobrazovanie počtu prejdých krokov, prejdenú vzdialenosť v metroch a krivku reprezentujúcu zrýchlenie mobilného zariadenia v jednotkách ms^{-2} má na starosti trieda *GraphView*. Vzťahy medzi triedami balíku možno vidieť na obr. 5.1.2.

cz.muni.fi.pedometer.filters – balík pozostáva z tried tvoriacich Kálmánov filter, slúžiaci na filtráciu dát prúdiacich zo senzorov.

cz.muni.fi.pedometer.res – súčasťou tohto balíka sú triedy, ktoré zabezpečujú rôzne matematické výpočty prebiehajúce na maticiach. Trieda *Matrix* spracúva maticové výpočty (pozri 3.1 Jama). V balíku sa nachádzajú aj triedy reprezentujúce dekompozície. Tieto triedy využíva trieda *Matrix* na rozklady matic podľa rozličných princípov.



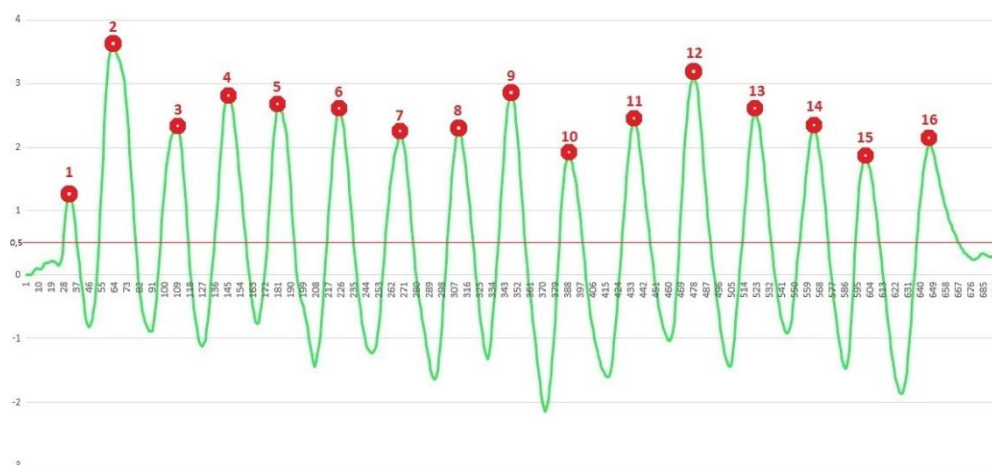
Obr. 5.1.1 Vzťahy medzi balíkmi

Obr. 5.1.2 Vzťahy medzi triedami balíku *cz.muni.fi.pedometer.main*

5.2 Detekcia krokov

Pri detekcii krokov ide o rozpoznanie náhlych zmien v strednej úrovni akcelerácie. Počas kráčania s mobilným zariadením v ruke dochádza k zaznamenávaniu zrýchlenia senzormi. Pohyb vpred spôsobí prudké pozitívne vychýlenie zrýchlenia, no po každom kroku dochádza tiež k prudkému zníženiu hodnoty zrýchlenia. Hodnoty zrýchlenia tak vytvárajú krivku znázornenú na obr. 5.2.1. Problémom pri detekcii krokov môže byť šum obsiahnutý v signáli. Ak je prejdený krok naozaj malý, môže byť šumom prekrytý a algoritmus krok nerozpozna. Preto pred samotným rozpoznávaním krokov musí byť signál vyfiltrovaný príslušnými filtrami (4.3 Filtrácia dát).

Každý krok je započítaný po splnení určitých podmienok. Aby boli eliminované rôzne vychýlenia spôsobené trasením rúk, má detekčný algoritmus stanovenú hranicu zrýchlenia. Konkrétne zariadenie musí prekonať hodnotu zrýchlenia $0,5ms^{-2}$. Pri detekcii tiež treba uvážiť, že prejdenie jedného kroku musí byť ohraničené aj časovými podmienkami. Maximálna časová dĺžka kroku je stanovená na dve sekundy, minimálna na 0,6 sekundy. Na obr. 5.2.1 sú znázornené miesta, kde algoritmus rozpoznal 16 krokov. Červená horizontálna čiara znázorňuje minimálnu hodnotu zrýchlenia.



Obr. 5.2.1 Graf zrýchlenia v čase (v ms^{-2}) s vyznačenými krokmi

5.3 Určenie prejdenej vzdialenosti

Okrem detekovania krokov je úlohou aplikácie spočítať prejdenu vzdialenosť v metroch. Toho sa dá dosiahnuť viacerými spôsobmi. Prvým je dvojný integrál zrýchlenia cez čas. Integrovaním zrýchlenia sa spočíta rýchlosť, následne ďalšie integrovanie rýchlosti produkuje prejdenu vzdialenosť v metroch. Druhý spôsob počíta prejdenu vzdialenosť súčinom počtu detekovaných krokov a predom stanovenej dĺžky jedného kroku. Obidva spôsoby majú svoje výhody aj nevýhody, a taktiež oba spôsoby nie sú stopercentne presné.

5.3.1 Výpočet vzdialenosti integrálom

K výpočtu prejdenej vzdialenosti integrálom sú potrebné viaceré parametre. Integrovanou hodnotou je akcelerácia zariadenia očistená od šumov a chýb. Zrýchlenie sa integruje cez čas, teda sú dôležité aj časové údaje, konkrétne čas aktuálnej vzorky zrýchlenia. Vzorec na výpočet :

$$\Delta x = \int_{t_0}^t v_x(t) dt = \int_{t_0}^t \left[\int_{t_0}^{t'} a_x(t') dt' \right] dt \quad (3)$$

Samotný integrál ale nie je možné algoritmicky reprezentovať. Preto je na implementáciu takéhoto vzorca potrebné lichobežníkové pravidlo, ktoré hodnotu integrálu aspoň priblíži. Aproximácia integrálu lichobežníkovým pravidlom spočíva v tom, že dva susedné body sú spojené v lichobežníkový útvar a následne sa spočíta obsah vzniknutého útvaru. Vzorec pre lichobežníkové pravidlo :

$$\int_a^b f(x) dx \approx \frac{b-a}{2} (f(a) + f(b)) \quad (4)$$

$$\Delta v \approx \frac{t-t_0}{2} (a(t) + a(t_0)) \quad (5)$$

$$\Delta x \approx \frac{t-t_0}{2} (v(t) + v(t_0)) \quad (6)$$

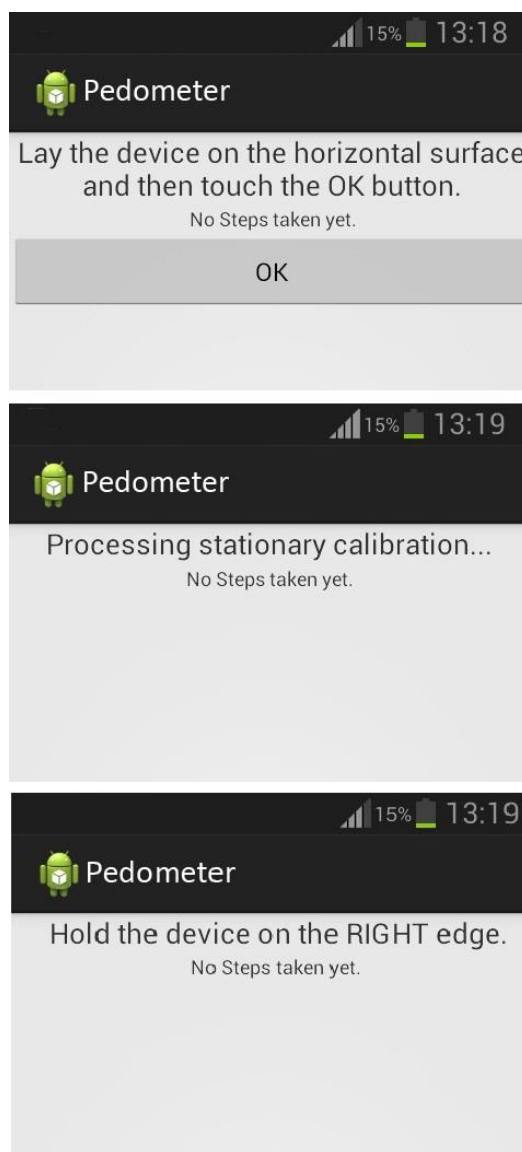
S výpočtom prejdenej vzdialenosti integrálom je spojených viacero problémov. Keďže filtráciou nemožno dokonale eliminovať všetky problémy, integrálom sa zväčšujú a prehlbujú chyby signálu, ktoré pri filtrácii neboli odstránené. Simultánnym integrovaním zrýchlenia za behu programu sa tak kumulujú zostatkové chyby a šumy, čo spôsobuje výrazné odchýlenia od reality.

5.3.2 Fixná dĺžka kroku

Výpočet prejdenej vzdialenosti fixnou dĺžkou kroku vyžaduje dva parametre. Počet prejdenej krokov a dĺžku jedného kroku. Dĺžka jedného kroku sa stanoví pred samotným behom aplikácie. Aby sa čo najviac zamedzilo nepresnostiam, dĺžka kroku sa spočíta ako podiel fyzicky prejdenej vzdialenosti a počtu krokov. Vhodné by bolo prejsť aspoň 10 krokov a následne prejdenu vzdialenosť vydeliť počtom prejdenej krokov. Výsledná dĺžka kroku vyjadruje priemernú dĺžku kroku na prejdenej vzdialenosti. Nevýhod tejto metódy je viacero. Pri samotnom meraní dĺžky kroku dochádza k odchýlke a nepresnostiam. Pri meraní prejdenej vzdialenosti sa dĺžka každého kroku líši, a preto v konečnom dôsledku je výsledok v metroch iba približný. Odchýlka narastá spolu s prejdenu vzdialenosťou.

5.4 Grafické prostredie

Po spustení aplikácie je užívateľ vyzvaný na stlačenie tlačidla *OK* po ktorom sa spustí statická kalibrácia. Po ukončení procesu statickej kalibrácie nasleduje kalibrácia dynamická a teda je nutné telefón podľa výziev natáčať jednotlivými hranami kolmo k zemi. Následne sú údaje o kalibrácii uložené a program môže prejsť do stavu vzorkovania.



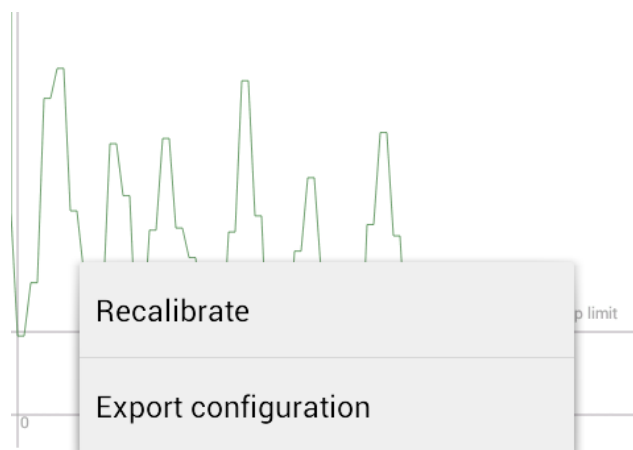
Obr. 5.4.1 Hlásenia zobrazované po spustení aplikácie

Po vykonaní potrebných kalibrácií nasleduje samotné prostredie programu. Graf zobrazuje zmenu zrýchlenia v čase. V grafe možno vidieť hranicu, ktorú musí zrýchlenie prekročiť, aby bol detekovaný krok. Nad grafom sa nachádzajú informácie o prejdenej vzdialenosti. Vzdialenosť vypočítaná dvojným integrálom zrýchlenia cez čas sa nachádza vľavo, vzdialenosť vypočítaná súčinom počtu krokov a dĺžky jedného kroku sa nachádza v strede. Na pravej strane sa nachádza údaj o aktuálnom zrýchlení zariadenia. Táto hodnota je vykresľovaná grafom. V hlavičke zobrazenia sa nachádzajú informácie o stave aplikácie a počte prejdenej krokov.



Obr. 5.4.2 Grafická reprezentácia dát aplikácie

Pri opakovanom spustení aplikácie program načíta uložené dáta o kalibrácii a spustí vzorkovanie. Ak je však nevyhnutné kalibráciu z rôznych príčin opakovať, dá sa tak vykonať po stlačení tlačidla *Nastavení*. Po stlačení tlačidla nastavení sa zobrazí v dolnej časti obrazovky kontextová ponuka s možnosťami rekalibrácie a exportovania konfigurovaných kalibračných dát. Po stlačení tlačidla *prekalibrovať* aplikácia spustí opätovnú kalibráciu dát (Obr. 5.4.1) a po jej prebehnutí pokračuje vzorkovaním (Obr.5.4.2). Po stlačení tlačidla *exportovať konfiguráciu* sa aktuálne kalibračné dáta uložia na pamäťové médium v podobe textového dokumentu.



Obr. 5.4.3 Kontextová ponuka rekalibrácie a exportovania dát

6 Testovanie

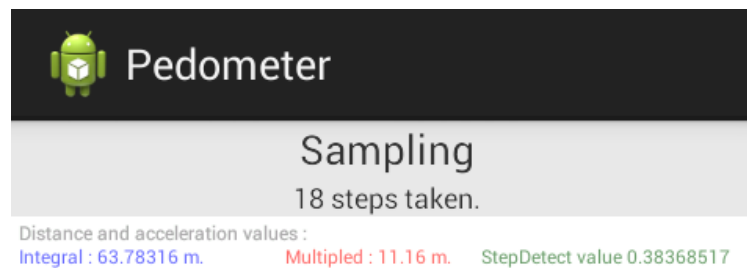
Proces implementácie strieda proces testovania. Na implementovanú aplikáciu je potrebné otestovať, aby bolo možné určiť s akou presnosťou algoritmus dokáže spočítať počet krokov a prejdenu vzdialenosť. Zariadenie bolo počas testovania držané v ruke, vo zvislej orientácii kolmo k zemi.

Testovanie behu aplikácie sa realizuje na zariadení Samsung Galaxy SIII mini (GT 18190N). Výbavou prístroja je trojosový senzor magnetického poľa, trojosový akcelerometer a trojosový gyroskop. Bližšie špecifiká senzorov získané z aplikácie Android Hardware Info sú obsiahnuté v nasledujúcej tabuľke 6.1:

názov senzora	výrobca	rozsah	presnosť	oneskorenie
APLS 3-axis Magnetic field sensor	Alps electric co., ltd.	4800 μT	0,149 μT	10 ms
MPU-6050 Accelerometer	Invensense	39,24 $\frac{m}{s^2}$	0,153 $\frac{m}{s^2}$	10 ms
MPU-6050 Gyroscope	Invensense	2000,01 $\frac{rad}{s}$	0,061 $\frac{rad}{s}$	10 ms

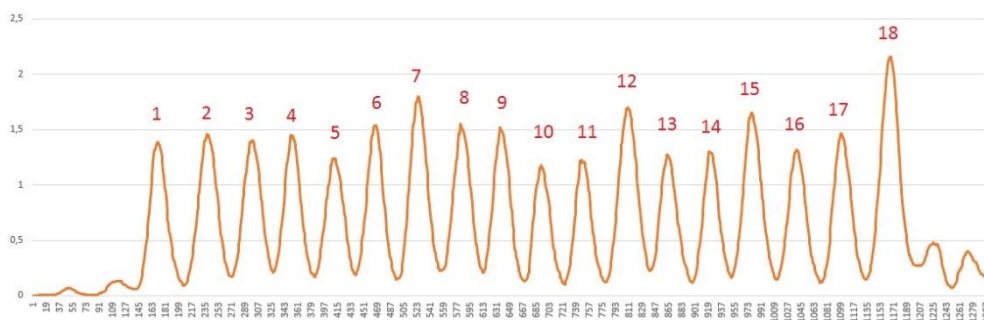
Tabuľka 6.1 Vlastnosti senzorov zariadenia

Procesom testovania sa do externého súboru (formát .csv) ukladaného na pamäťové médium zaznamenávajú informácie o zrýchlení, prejdenej vzdialenosti spočítanej dvojným integrálom zrýchlenia a prejdenej vzdialenosti spočítanej súčinom počtu krokov a fixne stanovenej dĺžky jedného kroku. Z týchto informácií je následne možné zostaviť jednotlivé grafy, ktoré majú výslednú výpovednú hodnotu.



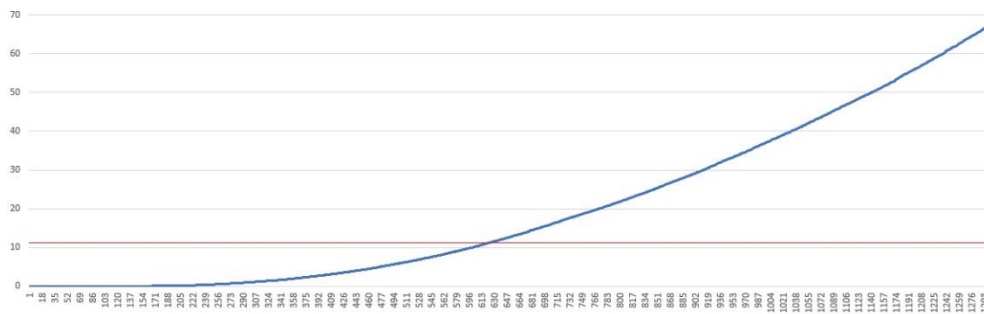
Obr. 6.1 Informácia zobrazovaná po vykonaní testu

Pri testovaní zariadenie prekonal fyzicky odmeraných 11 metrov. Na jedenástich metroch, ako možno vidieť na obr. 6.2, bolo zaznamenaných celkom 18 ostrých vychýlení zrýchlenia, ktoré možno považovať za vychýlenia spôsobené krokmi. Tento údaj sa zhoduje s počtom detekovaných krokov algoritmom aplikácie (viď obr. 6.1).

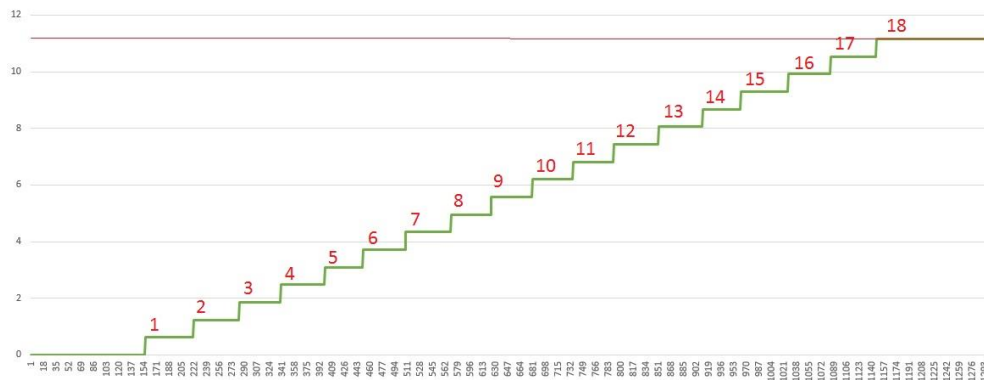


Obr. 6.2 Vychýlenia zrýchlenia (v ms^{-2}) spôsobené krokmi užívateľa

Spolu so zaznamenávaním zrýchlenia dochádza k výpočtu prejdenej vzdialenosti dvoma spôsobmi. Na obr. 6.3 je znázornený výsledok výpočtu prejdenej vzdialenosti dvojným integrálom zrýchlenia. Obr. 6.4 reprezentuje výpočet prejdenej vzdialenosti súčinom, zobrazené a číselne označené sú aj prejdené kroky. Horizontálna červená čiara na oboch obrázkoch znázorňuje reálne prejdenú vzdialenosť 11 metrov.



Obr. 6.3 Graf vzdialenosti v metroch vypočítanej integrálom



Obr. 6.4 Graf vzdialenosti v metroch vypočítanej súčinom

Výsledok výpočtu integrálom je výrazne odlišný od reálne prejdenej vzdialenosti. Aplikácia zobrazuje, že zariadenie prekonalo 63,78 metra. Výsledok je pravdepodobne ovplyvnený chybami, ako sú šum, trasenie rúk a odchýlka merania, ktoré integrovaním cez čas ešte väčšmi prehľbujú vychýlenie od reality.

Výsledok výpočtu súčinom sa tak výrazne od reality nelíši. Fixná dĺžka kroku bola stanovená na 0,62m. Po detekovaní 18 krokov aplikácia informuje, že zariadenie prekonalo 11,16m. Došlo teda k vychýleniu 16 cm od reálne prejdenej vzdialenosti. Takáto chyba sa s počtom prejdenej krokov taktiež zväčšuje, no chyba narastá lineárne, na rozdiel od predchádzajúceho prípadu. Tam narastá chyba exponenciálne. Meranie prejdenej vzdialenosti súčinom je tak výrazne presnejšie ako meranie využívajúce dvojný integrál.

7 Záver

Hlavným zámerom tejto práce bolo vytvorenie aplikácie, krokomeru, ktorý je určený pre mobilné zariadenia pracujúce na platforme Android. Aplikácia využíva vstavané senzory, konkrétne akcelerometer, gyroskop a magnetometer, na meranie zrýchlenia, počtu prejdých krokov a prejdenej vzdialenosti v metroch. Krokomer je primárne určený na ďalšie spracovanie, keďže sa má stať súčasťou inerciálneho navigačného systému. Okrem zapracovania do navigačného systému môže byť aplikácia používaná aj v aktuálnom stave, napríklad na meranie prejdenej vzdialenosti bežcov počas športovania.

Pred samotnou implementáciou bolo nutné sa vysporiadať s viacerými problémami. Zaznamenávanie údajov zo senzorov je značne ovplyvnené chybami, ktoré sa vyskytujú v každom z používaných signálov. Dochádza ku kumulovaniu šumov, náhodných a systematických chýb, chýb integrovania či k časovému oneskoreniu. Šumy dokáže algoritmus eliminovať fúziou senzorov, teda funkčným využívaním výhod a minimalizáciou nevýhod pri čerpaní dát prúdiacich z jednotlivých senzorov. Vplyv systematickej chyby na chod programu je odstránený hneď po spustení aplikácie a prebehnutí kalibrácie senzorov zariadenia. Ostatné chyby a vychýlenia sa snažia eliminovať Kálmánov a Low-Pass filter. Napriek tomu nie je možné všetky chyby a vplyvy prostredia eliminovať stopercentne, čo sa odzrkadľuje pri meraní prejdenej vzdialenosti.

K detekcií krokov dochádza pri pohybe zariadenia s cyklickým nárastom a poklesom zrýchlenia, čo je typické pre pohyb vznikajúci pri kráčaní. Vždy po splnení všetkých podmienok nutných k rozpoznaní kroku sa zvýši počítadlo prejdých krokov o jedna.

Simultánne s detekciou krokov dochádza k výpočtu prejdenej vzdialenosti. Výpočet je realizovaný dvojako – dvojným integrálom zrýchlenia cez čas a súčinom fixnej dĺžky kroku s počtom prejdých krokov. Ako preukázalo testovanie aplikácie, určovanie prejdenej

vzdialenosti dvojným integrálom je veľmi nepresné a pre ďalšie použitie nevhodné. Vhodnejším spôsobom merania prejdenej vzdialenosti sa ukázal byť súčin počtu prejdenej krokov s vopred stanovenou dĺžkou jedného kroku. Takéto meranie by s použitím ďalších podporných techník mohlo byť vhodné aj pri inerciálnom navigačnom systéme. Podporným systémom by sa mohla stať lokalizácia zariadenia prostredníctvom bezdrôtových sietí. Konkrétne pravidelná synchronizácia prejdenej vzdialenosti krokomeru s Wi-Fi lokalizáciou by mohli eliminovať odchýlky od reality a ponúknuť relatívne presné určenie polohy prístroja v rámci budovy. Preto by sa synchronizácia krokomeru s Wi-Fi lokalizáciou mohla stať predmetom ďalšieho skúmania.

Literatúra

- [1] MILETTE, Greg. *Professional Android sensor programming*. Indianapolis: Wiley, c2012. 517 p. ISBN 978-1-118-18348-9.

- [2] REUTER, Claus. *The V2, and the Russian and American Rocket Program*. Canada: German Canadian Museum of Applied History, 2012. ISBN 978-1894643054.

- [3] CHEN, Yongguang a Hisashi KOBAYASHI. *Signal Strength Based Indoor Geolocation*. In: [online]. [cit. 2014-03-20]. Dostupné z: <http://hisashikobayashi.com/papers/Wireless%20Geolocation%20Algorithms%20and%20Analysis/Signal%20Strength%20Based%20Indoor%20Geolocation.pdf>

- [4] LINK, Jó Ágila Bitsch. *FootPath: Accurate Map-based Indoor Navigation Using Smartphones* [online]. Piscataway, NJ: IEEE, 2011 [cit. 2014-03-20]. ISBN 978-145-7718-052. Dostupné z: <http://www.comsys.rwth-aachen.de/fileadmin/papers/2011/2011-IPIN-bitsch-footpath-long.pdf>.

- [5] KING, A. D. *Inertial Navigation – Forty Years of Evolution*. In: [online]. [cit. 2014-03-20]. Dostupné z: http://www.imar-navigation.de/downloads/papers/inertial_navigation_introduction.pdf

- [6] WEISSTEIN. *Rectangle Function*. *MathWorld - A Wolfram Web Resource* [online]. [cit. 2014-03-31]. Dostupné z: <http://mathworld.wolfram.com/RectangleFunction.html>

- [7] SINGHAL, Toshak, HARIT a VISHWAKARMA. *Kalman Filter Implementation on an Accelerometer sensor data for three state estimation of a dynamic system*. ISSN 2277 – 4378. In: [online]. [cit. 2014-03-31]. Dostupné z: <http://psrcentre.org/images/extraimages/IJRET016069.pdf>
- [8] WELCH, Greg a Gary BISHOP. An Introduction to the Kalman Filter. In: [online]. [cit. 2014-04-01]. Dostupné z: http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf

Zoznam obrázkov

4.3.2.1	Reprezentácia vstupných a výstupných dát	13
5.1.1	Vzťahy medzi balíkmi	15
5.1.2	Vzťahy medzi triedami balíku	15
5.2.1	Graf zrýchlenia v čase s vyznačenými krokmi	16
5.4.1	Hlásenia zobrazované po spustení aplikácie	19
5.4.2	Grafická reprezentácia dát aplikácie	20
5.4.3	kontextová ponuka recalibrácie a exportovania dát	21
6.1	Informácia zobrazovaná po vykonaní testu	23
6.2	Vychýlenia zrýchlenia spôsobené krokmi užívateľa	23
6.3	Graf vzdialenosti v metroch vypočítanej integrálom	24
6.4	Graf vzdialenosti v metroch vypočítanej súčinom	24