

11. naloga - Optimalno filtriranje

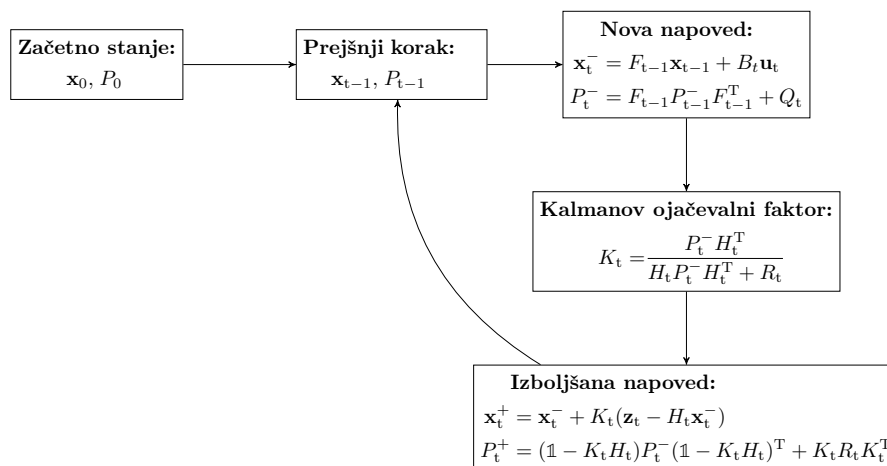
Tina Klobas

12. avgust 2020

1 Opis problema

Naj bo vektor \mathbf{x}_t vektor stanja ob času t , P_t pa pripadajoča kovariančna matrika, ki opisuje njegovo statistično negotovost. Potrebujemo še začetno stanje \mathbf{x}_0^+ ter kovarianco P_0^+ , ki ju dobimo iz meritve.

Potek Kalmanovega filtra je predstavljen na sledečem diagramu:



Vidimo, da je sestavljen iz treh pomembnih korakov:

1. Napovemo novo stanje ob času t ; \mathbf{x}_t^- in P_t^- . Matrika F_t je prehodna matrika sistema, ki jo dobimo s poznavanjem dinamike sistema. Iz gibalnih enačb dobimo tudi kontrolni vektor $\mathbf{c}_t = B_t \mathbf{u}_t$. Kovariančna matrika Q_t pa je posledica zašumljenosti podatkov.
2. Izračunamo *Kalmanov ojačevalni faktor*, ki določa s kolikšno utežjo v naslednjem koraku nova meritev prispeva k popravku. V enačbi nastopata matriki H_t – senzorsko okno (velikost odvisna od števila senzorjev in dimenzije \mathbf{x} in R_t , ki je kovariančna matrika šuma meritve).
3. Izboljšamo napoved stanja. \mathbf{z}_t , ki nastopa v enačbah je vektor meritev – podatkov, ki jih želimo obdelati.

2 Rekonstrukcija lokacije z GPS

Iz datoteke `kalman_cartesian_data.dat` dobimo podatke ob času t , zašumljene meritve položaja $\mathbf{r}_t = (x_t, y_t)$ hitrosti $\mathbf{v} = (v_{x,t}, v_{y,t})$ ter pospeškov $\mathbf{a} = (a_{x,t}, a_{y,t})$.

V našem primeru je vektor stanja $\mathbf{x} = (x, y, v_x, v_y)$, kontrolni vektor $\mathbf{c} = (0, 0, a_x \Delta t, a_y \Delta t)$, prehodna matrika je konstantna in je oblike:

$$F = \begin{bmatrix} \mathbb{1}_{2 \times 2} & \mathbb{1}_{2 \times 2} \Delta t \\ \mathbb{0}_{2 \times 2} & \mathbb{1}_{2 \times 2} \end{bmatrix},$$

šum časovne evolucije sledi iz napake pospeška, $Q_n = \text{diag}(0, 0, \sigma_a^2 \Delta t, \sigma_a^2 \Delta t)$, šum meritev pa iz napak GPS podatkov ter hitrosti, $R_n = \text{diag}(\sigma_{xy}^2, \sigma_{xy}^2, \sigma_v^2, \sigma_v^2)$.

Podatki so vzorčeni vsakih $\Delta t = 1,783$ s. Za pospeške in GPS podatke sta znani absolutni napaki $\sigma_{xy} = 25$ m in $\sigma_a = 0.05 \text{ m s}^{-2}$, za hitrost pa poznamo relativno napako: $\sigma_v = 0.01 \|\mathbf{v}\|$, pri čemer napako vseeno navzdol omejimo na 1 km h^{-1} .

Napovedati želimo trajektorijo vožnje in časovno odvisnost hitrosti, v primeru, da zmanjšamo gostoto vzorčenja – imamo na voljo le vsako peto meritev hitrosti in vsako deseto meritev lokacije. Zaradi zadnjega napotka, da ne dobimo na vsakem koraku nove meritve, moramo uporabiti tri različne okenske matrike H :

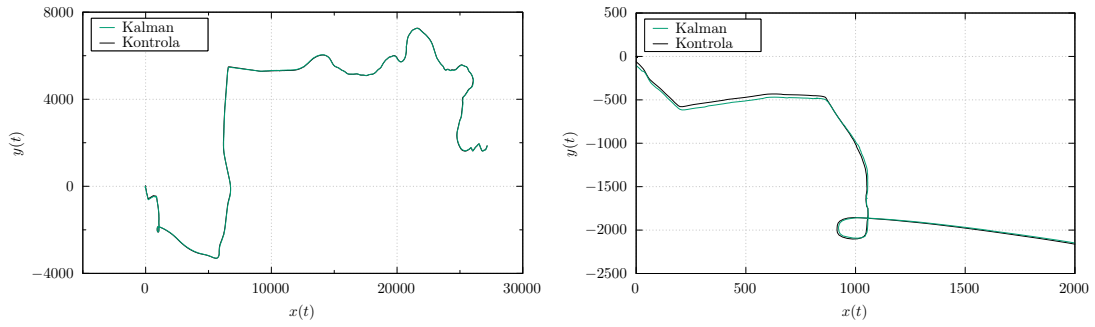
1. ne dobimo ne meritve \mathbf{v} ne meritve \mathbf{r} : $H_1 = \mathbb{0}_{4 \times 4}$,
2. dobimo le meritev \mathbf{v} : $H_2 = \begin{bmatrix} \mathbb{0}_{2 \times 2} & \mathbb{0}_{2 \times 2} \\ \mathbb{0}_{2 \times 2} & \mathbb{1}_{2 \times 2} \end{bmatrix}$,
3. dobimo obe meritvi: $H_3 = \mathbb{1}_{4 \times 4}$.

2.1 Rezultati

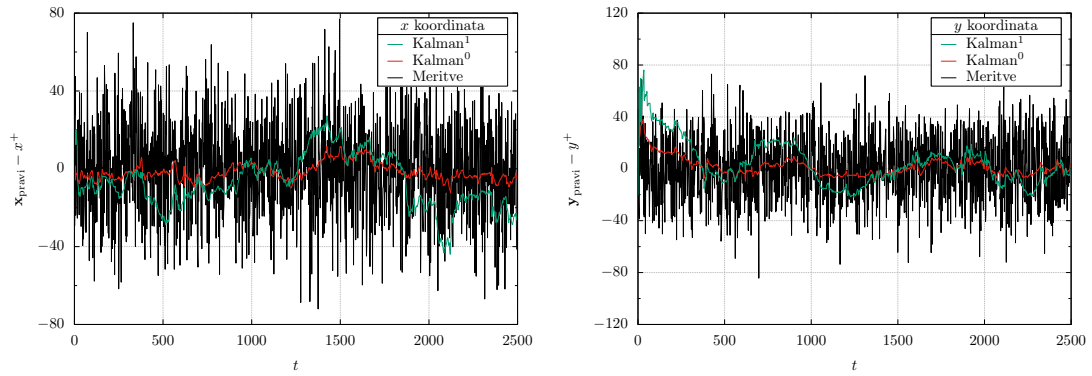
Rezultate uporabe Kalmanovega filtra na naših podatkih, lahko primerjamo s priloženo datoteko točnih vrednosti `kalman_cartesian_kontrola.dat`.

2.1.1 Točnost rezultatov

Na grafu 2.1 je prikazana dejanska pot in tista, ki smo jo dobili s Kalmanovim filtrom. Na približani sliki vidimo, da s Kalmanovim filtrom dobimo neko odstopanje, ki si ga oglejmo še na naslednjih grafih. Na sliki 2.2 so prikazani odmiki x koordinate pozicije od dejanskih vrednosti. Če primerjamo črn del z barvnima vidimo, da se s Kalmanovim filtrom res zelo izboljša natančnost meritve. S tem ko določamo katere meritve bomo na posamezne koraku vzeli (spreminjamo okensko funkcijo) tudi spreminjamo natančnost določanja lokacije, kar je tudi na tem grafu dobro opazno.

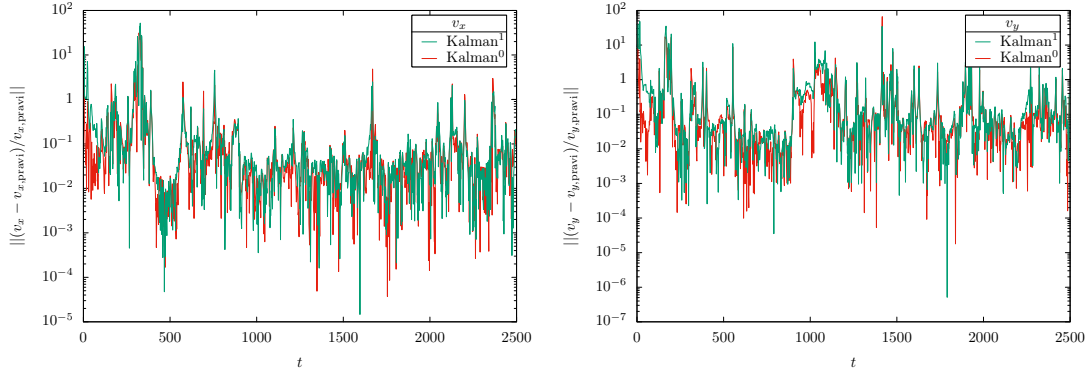


Slika 2.1: Dejanska pot vozila in njena rekonstrukcija s pomočjo Kalmanovega filtra. Na desni je prikazan izsek poti, kjer se bolje vidi odstopanje.



Slika 2.2: Na grafu je narisano odstopanje pozicije (levo x , desno y) od dejanskih podatkov in sicer; s črno so narisani odmiki zašumljene meritve, z rdečo je narisana Kalmanov filter, ki smo ga naredili na vseh meritvah, z zeleno pa Kalmanov filter, ko smo meritve poti jemali na vsakem 10. koraku, meritve hitrosti pa na 5.

Naslednja grafa 2.3 prikazujeta odstopanje hitrosti pri obeh variantah Kalmanovega filtra. Vidimo, da je relativna napaka pri obeh komponentah zelo majhna, večji vrhovi pa nastanejo pri naglih zavojih na poti.



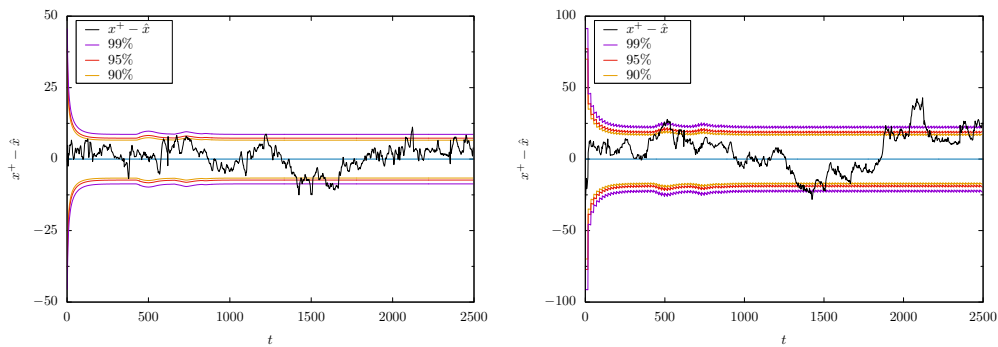
Slika 2.3: Na grafu je narisan odstopanje hitrosti (levo v_x , desno v_y) od dejanskih podatkov in sicer; z rdečo Kalmanov filter, ki smo ga naredili na vseh meritvah, z zeleno pa Kalmanov filter, kjer smo meritve poti jemali na vsakem 10. koraku, meritve hitrosti pa na 5.

Z matriko P :

$$P = \begin{bmatrix} \sigma_x^2 & \sigma_{x,y} & \sigma_{x,\dot{x}} & \sigma_{x,\dot{y}} \\ \sigma_{y,x} & \sigma_y^2 & \sigma_{y,\dot{x}} & \sigma_{y,\dot{y}} \\ \sigma_{\dot{x},x} & \sigma_{\dot{x},y} & \sigma_{\dot{x}}^2 & \sigma_{\dot{x},\dot{y}} \\ \sigma_{\dot{y},x} & \sigma_{\dot{y},y} & \sigma_{\dot{y},\dot{x}} & \sigma_{\dot{y}}^2 \end{bmatrix} \quad (1)$$

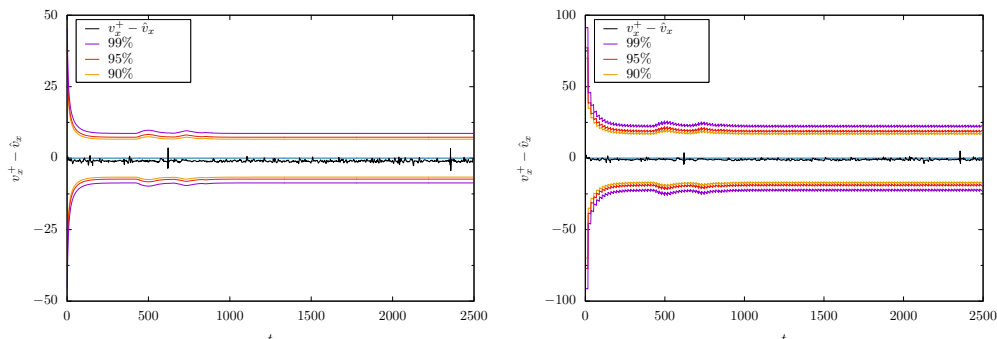
lahko določimo intervale zaupanja, oziroma elipsoide zaupanja. Izračunamo njene lastne vrednosti λ in ustrezne lastne vektorje \mathbf{q} , kar nam pomaga pri izračunu polosi: $a_i = \sqrt{s\lambda_i}$, in kot nagnjenosti elipse $\alpha_{ij} = \arctan \frac{\|\mathbf{q}_i\|}{\|\mathbf{q}_j\|}$.

Na grafu 2.4 sta narisana oba primera – na desni, kjer je prikazana rešitev za primer, ko imamo na voljo le vsako 10. meritev koordinate x vidimo, kako je potek radija intervalov nazobčan in če malo bolje pogledamo, lahko opazimo da je ta žagasta oblika dvojna; radij zaupanja se večja, dokler ne dobimo zopet nove meritve hitrosti in nato še večji preskok, ko dobimo novo meritev koordinate. Nekje pri $t = 600$ je viden malo večji preskok – kar je na grafu 2.1 ravno točka zavoja v bližini $(1000, -2000)$. Na tem mestu pade natančnost meritve, s čimer se poveča interval zaupanja. Podobna preskoka sta tudi okoli $t = 1400$ in $t = 2100$, obakrat smo v bližini večjih zavojev.



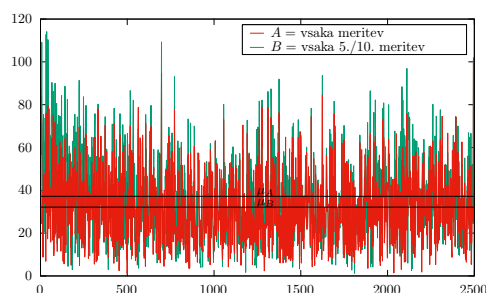
Slika 2.4: Odmik Kalmanove rešitve od prave vrednosti lokacije x , narisan med intervali zaupanja. Na levi strani smo vzeli vse meritve, na desni pa vsako 5. meritev hitrosti in vsako 10. meritev lokacije.

Podobno lahko naredimo tudi za meritev hitrosti v_x in dobimo graf 2.5, kjer opazimo, da pa naši rezultati s filtriranjem niso nikoli padli z izbranih intervalov zaupanja. Tudi tukaj vidimo enake preskoke, kot smo jih prej na grafih za koordinate.



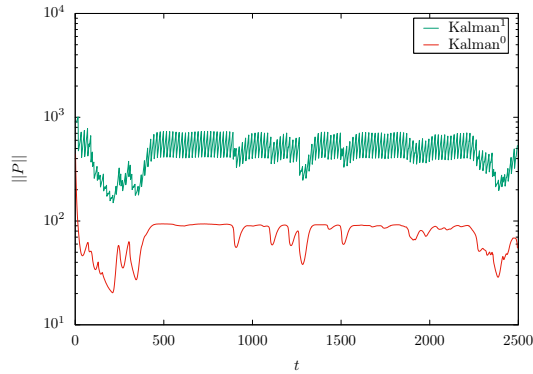
Slika 2.5: Odmik Kalmanove rešitve od prave vrednosti hitrosti v_x , narisan med intervali zaupanja. Na levi strani smo vzeli vse meritve, na desni pa vsako 5. meritev hitrosti in vsako 10. meritev lokacije.

Naslednja stvar, ki jo lahko tudi opazujemo so residuali $\|\mathbf{z}_t - H_t \mathbf{x}_t^-\|$. Njihovo spreminjanje s časom je prikazano na grafu 2.6, vendar nam k razumevanju sistema nič ne doprinese; rezultati so precej razpršeni, povprečno vrednost lahko izračunamo in jo tudi narišemo, vendar tega ob taki razpršeni rešitvi, ni koristno početi.



Slika 2.6: Potek 2. norme residuala v primeru, ko imamo na voljo vse meritve in, ko imamo samo vsako 5. meritev hitrosti in vsako 10. meritev lokacije.

Na naslednjem grafu 2.7 imamo prikazano normo kovariančne matrike P . Vidimo, da ko zmanjšamo pogostost vzorčenja povečamo negotovost stanja, poleg tega pa tudi tukaj opazimo to žagasto obliko – $\|P\|$ se zmanjša ob novi meritvi.



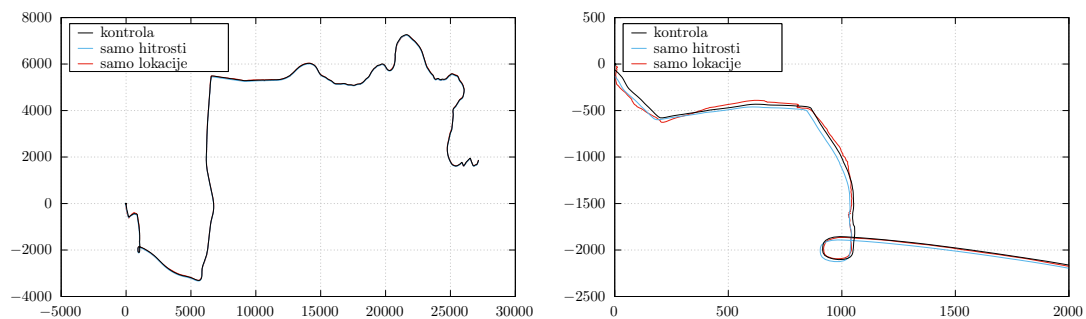
Slika 2.7: Norma kovariančne matrike P , kjer je z rdečo narisani filter pri katerem smo dobivali meritve na vsakem koraku, z zeleno pa smo vzeli lokacijo na vsakem 5., hitrost pa na vsakem 5.

3 Merjenje lokacije s telefonom

Problem je enak kot prej, le da imamo zdaj ob vsakem koraku dve možnosti:

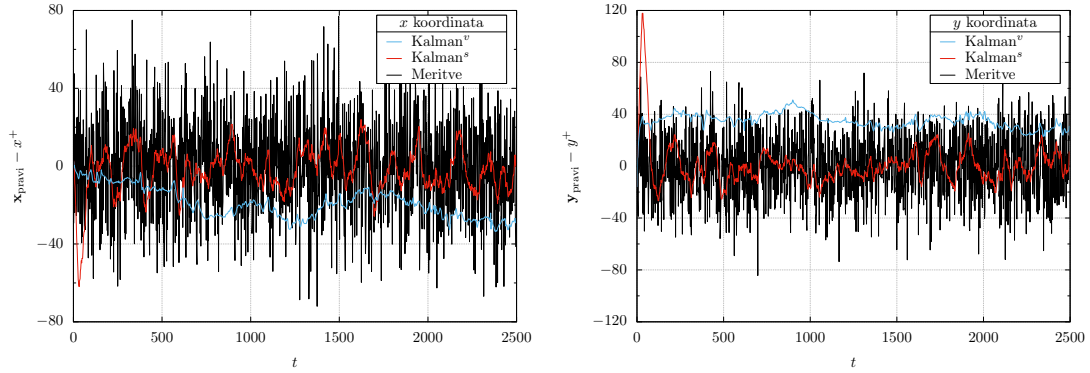
- (a) lahko merimo le lokacijo in pospeške,
- (b) ali pa lahko merimo le hitrosti in pospeške.

Tako kot v prvem delu naloge, lahko tudi zdaj narišemo trajektorije. Če dobljeni graf 3.1 primerjamo s tistim iz prvega dela 2.1, na prvi pogled ocenimo, da je metoda, kjer smo dobivali vse podatke, seveda bolj natančna kot ti dve in da je Kalmanov filter samo s podatki o legah bolj skače okoli prave vrednosti kot pa tisti, ki ima na voljo le hitrosti.



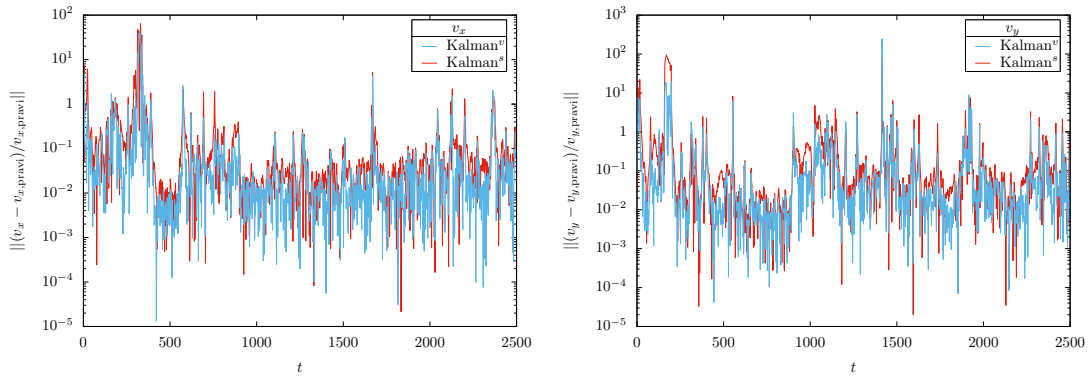
Slika 3.1: Primerjava poteka poti med točnim rezultatom, Kalmanovim filtrom, ko imamo na voljo samo podatke o pozicijah (z rdečo) in ko imamo na voljo samo meritve hitrosti (s svetlo modro). Na desni je približano območje okoli prvega krožnega zavoja, kjer se razlike med tremi potmi bolje razločijo.

Preverimo bolj natančno na grafih napak – na 3.2 vidimo, da filter z lokacijami res več skače okoli prave vrednosti, vendar je bližje pravi vrednosti, kar je veliko bolj očitno pri y koordinato. Tukaj zgleda, da je filter v začetku *vrglo* s tira in ker potem ni dobil nobenega podatka o točni lokaciji, popravke pa delal le na podlagi hitrosti, imel približno konstantno odstopanje od točne lege.



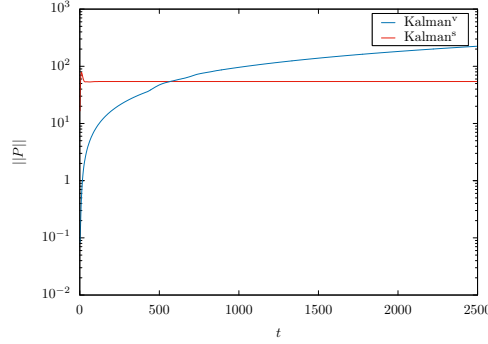
Slika 3.2: Odstopanje pozicije (x na levi in y na desni) od točnih podatkov. S črno je narisan zašumljena meritev, z rdečo meritev samo s pospeški in lokacijo in z modro samo s pospeški in hitrostmi.

Grafi odstopanj hitrosti 3.3 so podobni tistim iz prvega dela. Napaki za obe varianti filtra imata enako obliko – pri istih časih imamo maksimalne odstopanja hitrosti. To se zgodi v točkah ko imamo nagle spremembe hitrosti – filter potrebuje korak da ujame nazaj točno vrednost. Da se to zgodi v istih točkah niti ni presenetljivo, saj je ta nagla sprememba hitrosti v smeri posledica zavojev poti.



Slika 3.3: Odstopanje hitrosti (v_x na levi in v_y na desni) od točnih podatkov. Z rdečo je Kalmanov filter kjer dobivamo samo podatke o pospeških in lokacijah in z modro samo o pospeških in hitrostih.

Na grafu 3.4 norme kovariančne matrike pri prvi metodi kjer imamo na voljo meritvami leg, konstantna – popravek iz matrike šuma R je ves čas konstanten, saj nimamo člena z relativno napako σ_v . Pri drugi metodi, pa je ta člen z relativno napako razlog za to, da negotovost stanja vedno večja.



Slika 3.4: Norma matrike P – z rdečo je narisana Kalmanov filter kjer dobivamo samo podatke o pospeških in lokacijah in z modro samo o pospeških in hitrostih.

4 Orientacija vozila

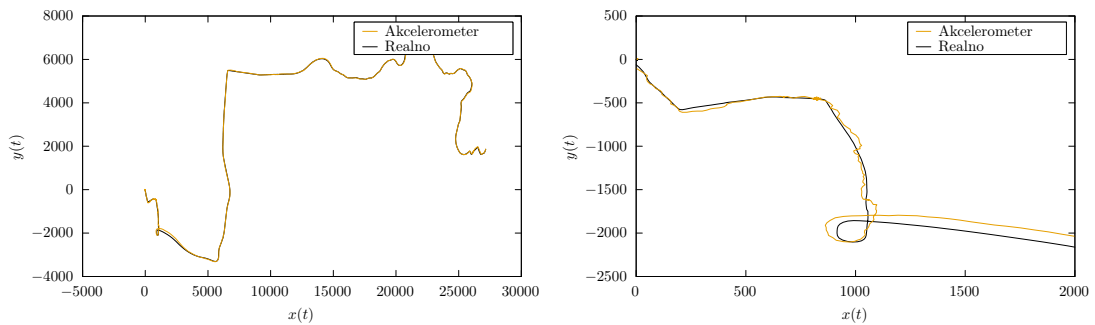
V resnici nam akcelerator podaja pospeške $\mathbf{a} = (a_t, a_r)$ glede na trenutno orientacijo vozila. Med kontrolnim vektorjem in nehomogenim delom dinamičnega modela zato stoji še ena linearna preslikava $B_n : \mathbf{c}_n = B_n \mathbf{u}_n$. V tem primeru gre za ortogonalno transformacijo, definirano s trenutno oceno hitrosti:

$$\mathbf{u}_n = (\mathbf{0}, \mathbf{a}_n \Delta t), \quad B_n = \begin{bmatrix} \mathbb{1}_{2 \times 2} & \mathbb{0}_{2 \times 2} \\ \mathbb{0}_{2 \times 2} & B_n^{vv} \end{bmatrix}, \quad B_n^{vv} = \frac{1}{\|\mathbf{v}_n\|} \begin{bmatrix} v_x & -v_y \\ v_y & v_x \end{bmatrix}. \quad (2)$$

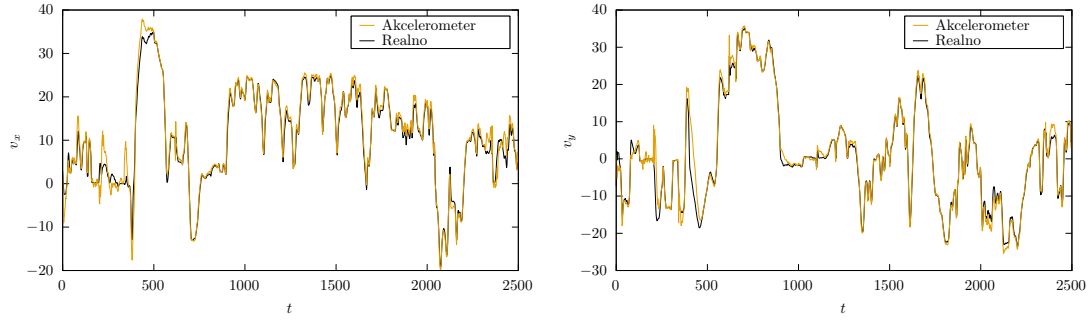
Upoštevati moramo še negotovost trenutne ocene hitrosti:

$$Q_n^{vv} = \Delta t^2 \left\{ \sigma_a^2 \mathbb{1}_{2 \times 2} + \frac{\mathbf{v}_n^\perp P_n^{vv} \mathbf{v}_n^\perp}{\|\mathbf{v}_n\|^4} [(B_n^{vv} \mathbf{a}_n^\perp) \otimes (B_n^{vv} \mathbf{a}_n^\perp)] \right\}, \quad (3)$$

kjer sta Q_n^{vv} in P_n^{vv} hitrostno-hitrostna bloka 4×4 kovariančnih matrik. Vektorja \mathbf{v}_n^\perp in \mathbf{a}_n^\perp sta hitrost in pospešek, zavrtena v pozitivni smeri za $\frac{\pi}{2}$. Pot bomo rekonstruirali iz danih podatkov `kalman_relative_data.dat`, katere stolpci so čas t , GPS položaj (x, y) ter pospeška (a_t, a_r) . Na grafu 4.1 je narisana dejanska pot in ta, ki smo jo dobili s Kalmanovim filtrom. Na prvi pogled nas tudi tukaj najbolj vrže s tira prvi zavoj. Na grafih hitrosti 4.2 vidimo, da tudi hitrost v tej točki najbolj zgrešimo.

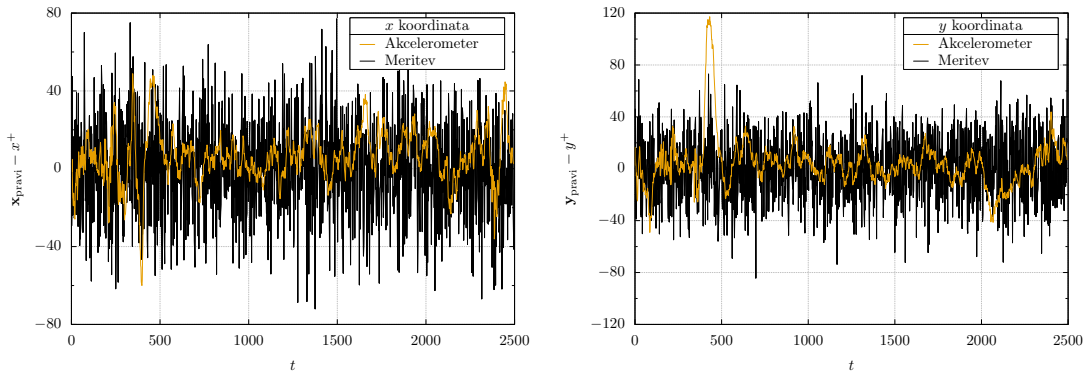


Slika 4.1: Pot vozila in njena rekonstrukcija s Kalmanovim filtrom, če dobivamo podatke o orientaciji vozila.

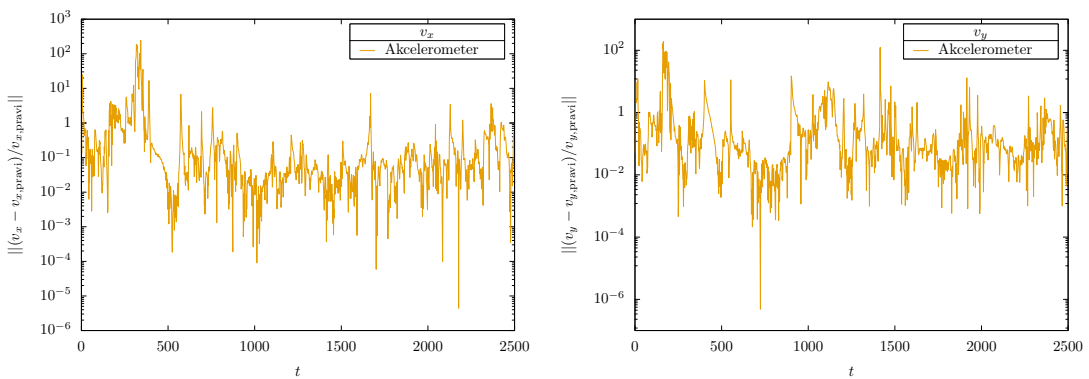


Slika 4.2: Hitrosti, ki jih dobimo s Kalmanovim filtrom ob upoštevanju orientacije vozila in prave vrednosti.

Grafi odstopanja lege 4.3 potrdijo, to kar smo opazilo – največji odmik od prave vrednosti je v prvem krožnem zavoj. Enako je tudi s hitrostmi 4.4.

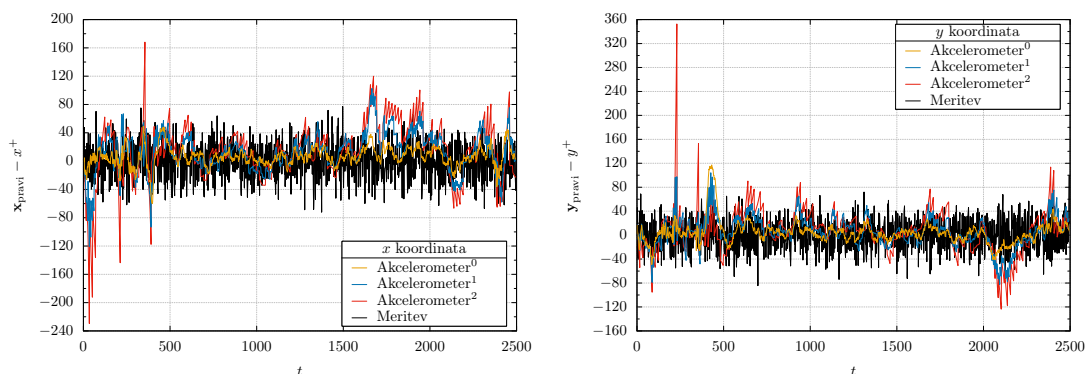


Slika 4.3: Pot vozila in njena rekonstrukcija s Kalmanovim filtrom, če dobivamo podatke o orientaciji vozila.

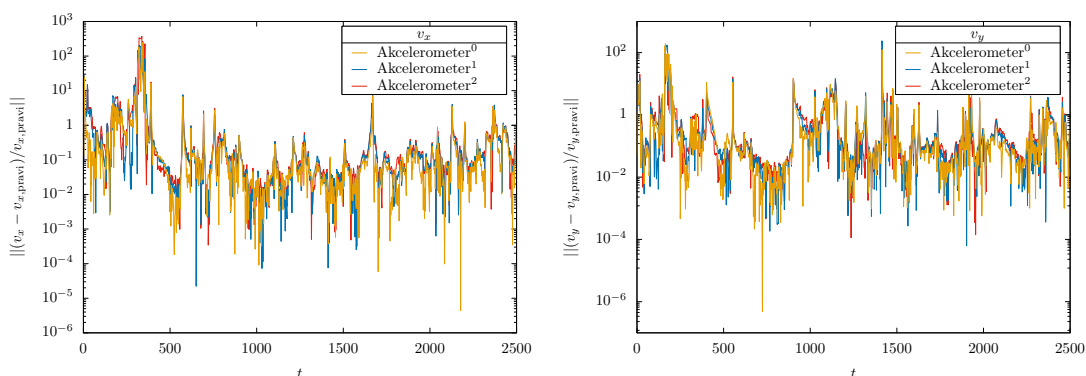


Slika 4.4: Pot vozila in njena rekonstrukcija s Kalmanovim filtrom, če dobivamo podatke o orientaciji vozila.

Tako kot v prvem delu naloge lahko pogledamo kako pogostost vzorčenja vpliva na natančnost Kalmanovega filtra. Na spodnjih grafih 4.5 in 4.6 vidimo podobno kot v prvem delu naloge – če zmanjšamo pogostost vzorčenja, povečamo odstopanje rešitve. Zanimivo je, da smo pri y koordinati v točki krožnega zavoja ($t \approx 450$) z manj pogostim vzorčenjem bližje rešitvi. Na grafu odmika hitrosti 4.6 opazimo, da smo tudi hitrost bolje zadeli z manjšim vzorčenjem. To, da smo v eni točki izboljšali natančnost, pa seveda ne odtehta tega, da smo jo drugje poslabšali. Lahko bi naredili kombinacijo in vzorčili na vsakem koraku, le na enem območju redkeje.

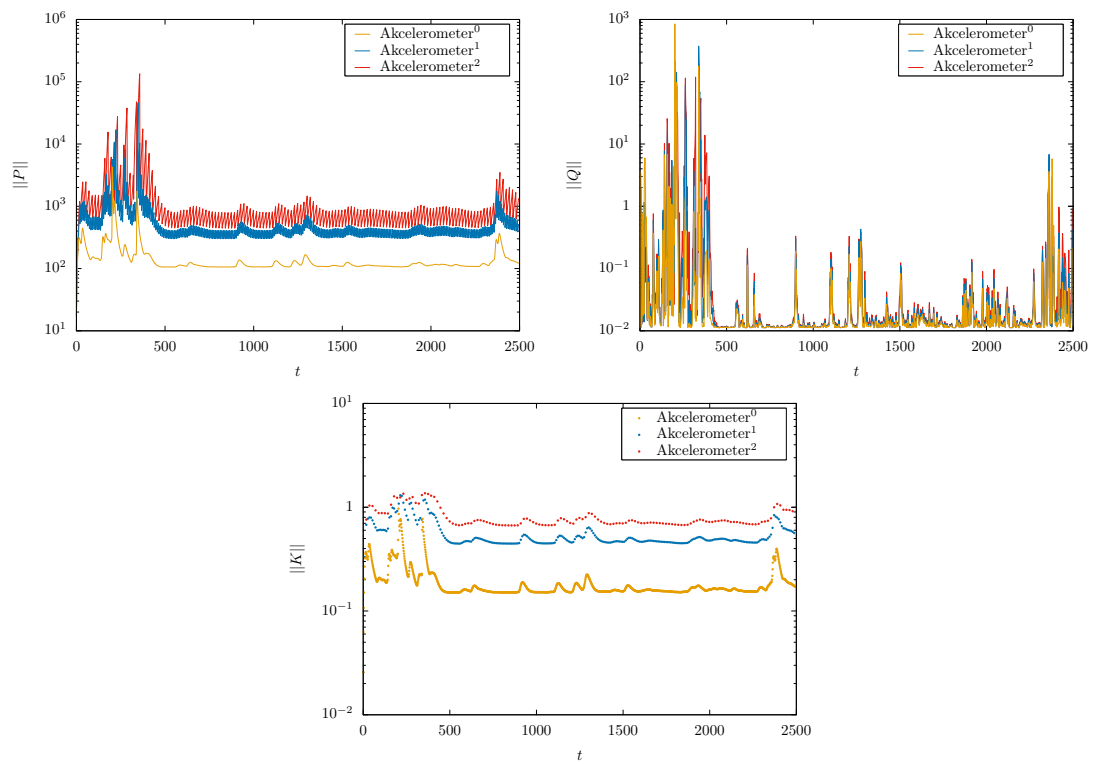


Slika 4.5: Odstopanje pozicije (x na levi in y na desni) od točnih podatkov. S črno je narisan zašumljena meritev, z oranžno smo meritve lokacije dobili na vsakem koraku, z modro na vsakem 5. in z rdečo na vsakem 10.



Slika 4.6: Odstopanje hitrosti (desno v_x , levo v_y) za Kalmanov filter, kjer dobivamo podatke o orientaciji. Z oranžno je narisan filter kjer smo meritve lokacije dobili na vsakem koraku, z modro na vsakem 5. in z rdečo na vsakem 10.

Poglejmo si še grafe 4.7, kjer so prikazane norme kovariančne matrike vektorja stanja P , kovariančne matrika vektorja šuma Q , in faktorja ojačanja K . Pri matriki P vidimo žagasto obliko, ki je posledica vzorčenja – po 5ih oziroma 10ih korakih se napaka spet izboljša. Velik Q dobimo v tistih točkah, ko je hitrost slab približek prave, še posebej v prvem delu meritve – do krožnega zavoja. Graf norme Kalmanovega filtra ima enako obliko kot graf norme P (saj sta linearno odvisna), poleg tega pa je redkejši pri manjšem vzorčenju, saj je povezan tudi z okensko funkcijo H , ki je 0 na korakih, ko ne dobivamo podatkov o meritvah.



Slika 4.7: Norme matrik P , Q in K .