

Univerza v Ljubljani
Fakulteta *za matematiko in fiziko*



Optimalno filtriranje

Avtor: Simon Perovnik

Predavatelj: prof. dr. Simon Širca

Asistent: doc. dr. Miha Mihovilovič

Enajsta naloga pri Modelski analizi I

Ljubljana, avgust 2022

1 Uvod

Kadar z zaporednimi meritvami spremljamo časovni razvoj procesa, katerega dinamiko poznamo, lahko s filtriranjem dosežemo občutno boljšo natančnost, kot jo dajejo surove meritve. Kalmanov filter deluje na principu optimalnega uteževanja linearne napovedi stanja ter novih meritev, kar doseže s sprotnim vodenjem evidence o kovarianci trenutne ocene stanja. Naj bo x_n vektor stanja sistema ob času t_n , P_n pa pripadajoča kovariančna matrika, ki opisuje njegovo statistično negotovost. Potrebujemo še začetno stanje x_0^+ ter kovarianco P_0^+ , ki ju običajno dobimo iz prve surove meritve. Komponentam stanja, ki niso na voljo, nastavimo velike začetne kovariance. Kalmanov filter poteka v dveh korakih. Prvi korak je izvedba časovnega koraka na trenutni napovedi stanja. Fizikalni sistem naj uboga časovno evolucijo $x_{n+1} = F_n x_n + c_n + w_n$, kjer je F_n prehodna matrika sistema, c_n kontrolni vektor, w_n pa vektor Gaussovega šuma s povprečjem 0 in kovariančno matriko Q_n . Napovemo novo stanje x^- ter njegovo kovariančno matriko P_{n+1}^- :

$$\begin{aligned} x_{n+1}^- &= F_n x_n^+ + c_n, \\ P_{n+1}^- &= F_n P_n^+ F_n^T + Q_n \end{aligned} \quad (1)$$

Sledi izboljšava te napovedi z novimi meritvami. V splošnem ne merimo neposredno komponent vektorja stanja x_n , temveč neko linearno kombinacijo $z_n = H_n x_n + r_n$, kjer je H_n matrika, ki določa, kaj merimo, r_n pa šum meritve s kovariančno matriko R_n . Matrika H_n je lahko singularna, če merimo manj spremenljivk, kot je dimenzija sistema.

$$\begin{aligned} K_{n+1} &= P_{n+1}^- H_{n+1}^T (H_{n+1} P_{n+1}^- H_{n+1}^T + R_{n+1})^{-1}, \\ x_{n+1}^+ &= x_{n+1}^- + K_{n+1} (z_{n+1} - H_{n+1} x_{n+1}^-) \\ P_{n+1}^+ &= (I - K_{n+1} H_{n+1}) P_{n+1}^- \end{aligned} \quad (2)$$

Prva enačba določa izboljšanje natančnosti stanja zaradi novih meritev, K_{n+1} je pa faktor ojačanja, ki določa, s kolikšno utežjo nova meritev prispeva k popravku.

2 Rekonstrukcija poti in hitrosti vozila iz GPS podatkov o lokaciji in hitrosti

2.1 Naloga

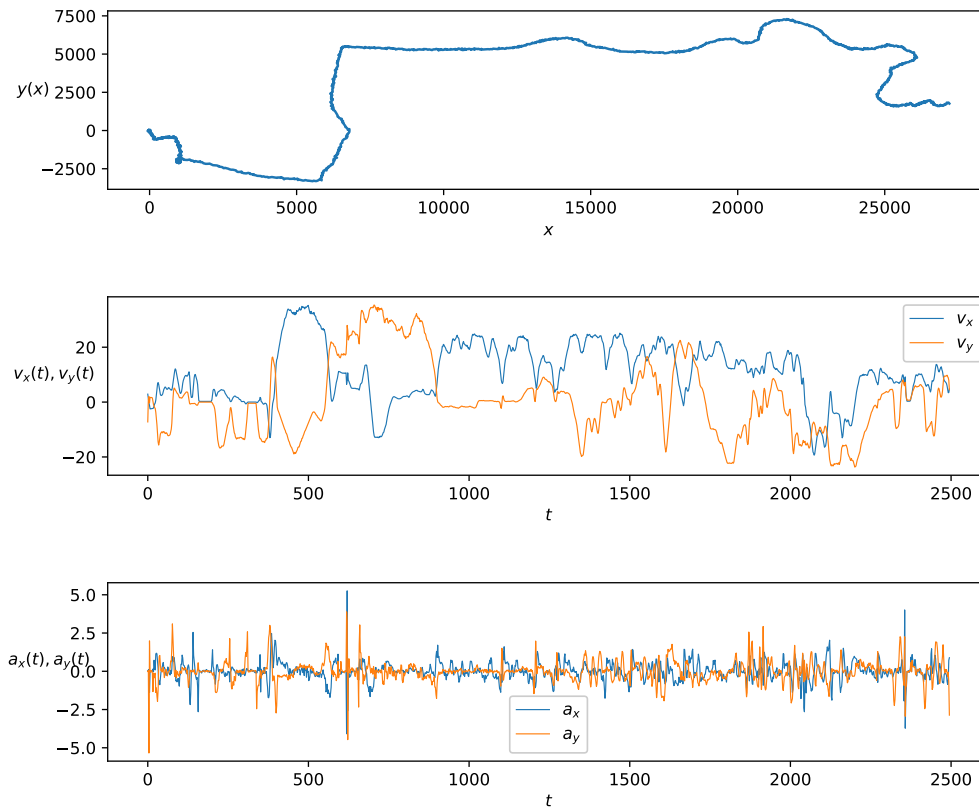
Tipičen primer uporabe Kalmanovega filtra je rekonstrukcija poti in hitrosti vozila na podlagi GPS podatkov o lokaciji, sprotnih podatkov o hitrosti vozila ter pospeškov z akcelerometra. V datoteki **kalman_cartesian_data.dat** so po stolpcih podani čas t , zašumljene meritve položaja (x_n, y_n) , hitrosti $(v_{x,n}, v_{y,n})$ ter pospeškov $(a_{x,n}, a_{y,n})$. Eksaktne vrednosti položajev in hitrosti za kontrolo pravilnosti najdete v datoteki **kalman_cartesian_kontrola.dat**. Vse količine so v osnovnih SI enotah. V našem primeru je vektor stanja $x = (x, y, v_x, v_y)$, kontrolni vektor dobimo iz pospeška, $c = (0, 0, a_x \Delta t, a_y \Delta t)$, prehodna matrika sistema je konstantna in izhaja iz kinematičnih zvez

$$F = \begin{bmatrix} I_{2 \times 2} & I_{2 \times 2} \Delta t \\ 0 & I_{2 \times 2} \end{bmatrix}, \quad (3)$$

šum časovne evolucije sledi iz napake pospeška, $Q_n = \text{diag}(0, 0, \sigma_a^2 \Delta t^2, \sigma_a^2 \Delta t^2)$, 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, $\sigma_a = 0.05 \text{ ms}^{-2}$, za hitrost pa poznamo relativno napako: $\sigma_v = 0.01 \|v\|$, pri čemer napako vseeno navzdol omejimo na 1 km/h. Napovej trajektorijo vožnje ter časovno odvisnost komponent hitrosti, v primeru, da vzorčenje ni tako gosto, kot je podano v datoteki, temveč dobimo le vsako peto meritev hitrosti in vsako deseto meritev lokacije. Katere meritve so na voljo, določate z matriko H_n , ki je v našem primeru vedno diagonalna. Kovarianca P_n vam pomaga pri določanju intervalov zaupanja za dobljene vrednosti, prav tako je koristno spremljati residue $\|z_{n+1} - H_{n+1} x_{n+1}^-\|$.

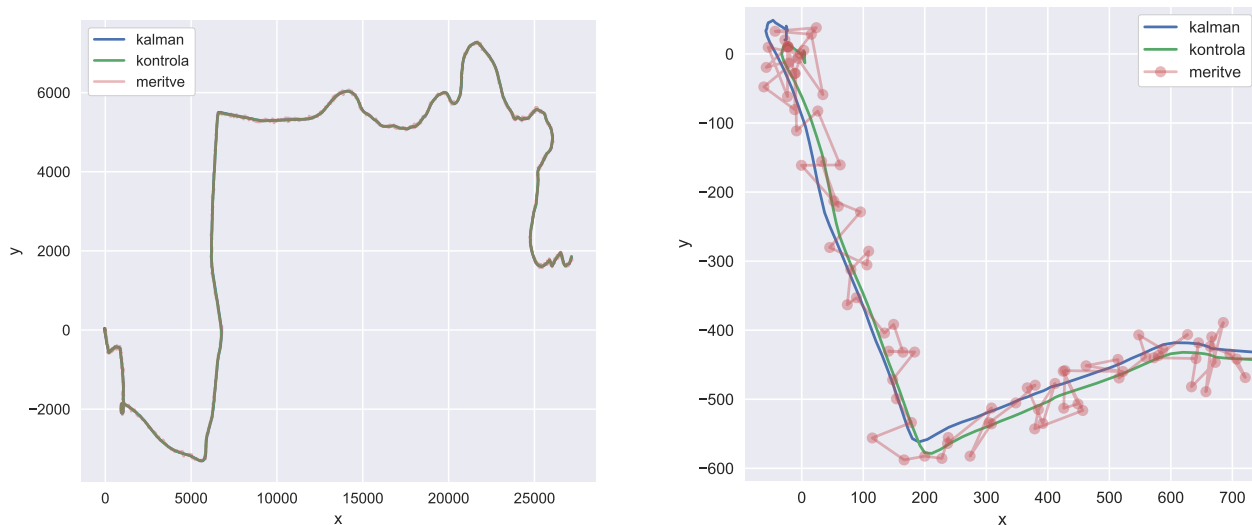
2.2 Polno vzorčenje hitrosti in položaja

Naloga nam podaja GPS podatke o lokaciji in hitrosti našega vozila med vožnjo, pravtako pa so nam na voljo podatki iz akcelerometra, ki jih bomo uporabili kot kontrolo.



Slika 1: Podatki iz datoteke **kalman_cartesian_data.dat**.

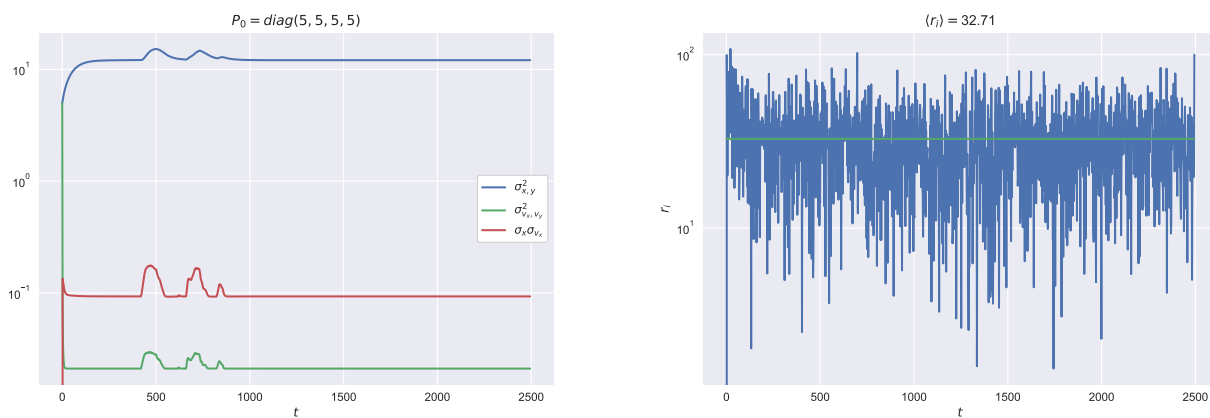
Preden si pogledamo, kako deluje naš filter v primeru redkejšega vzorčenja, si poglejmo, kakšno pot in hitrost napovemo ob upoštevanju vseh meritev. Pri tem bomo za začetno stanje x_0^+ vzeli kar vrednosti, ki nam jih poda prva meritev, za kovariančno matriko pa ocenimo $P_0^+ = \text{diag}(5, 5, 5, 5)$. Ker bomo uporabili vse podane podatke, bo matrika H kar $I_{4 \times 4}$.



Slika 2: Napovedana pot pri uporabi vseh podatkov. Na desni je približan začetni del poti.

- Pot se vsaj v grobem dobro ujema s podano kontrolo, nam pa 'pozoomirana' slika razkrije, da sledenje kljub vsemu ni povsem točno in hitro zgrešimo lokacijo za kakšnih 20 metrov.
- Znatne težave se pojavijo na začetku, ko se je avto najverjetneje počasi premikal po nekem parkirišču in so bili njegovi premiki na časovni interval meritev razmeroma majhni, glede na merilni šum.

Raziščemo lahko tudi, kako se tekom poti spreminjajo vrednosti znotraj kovariančne matrike in pa kaj se dogaja z residuali. Hitro ugotovimo, da je večino elementov kovariančne matrike pravzaprav enakih 0, neničelni ostanejo le $P_{11} = P_{22}, P_{33} = P_{44}, P_{13} = P_{31} = P_{42} = P_{24}$.

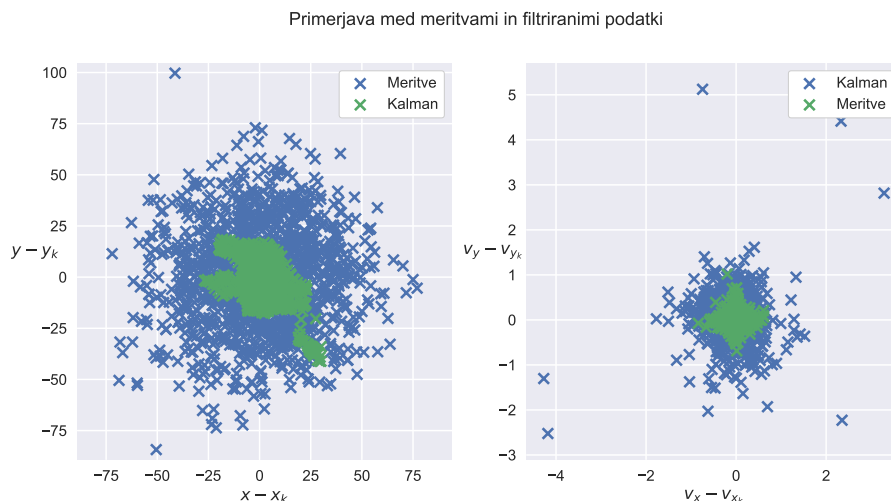


Slika 3: Elementi kovariančne matrike in residuali $\|z_{n+1} - H_{n+1}x_{n+1}^-\|$.

- Prvi residual je seveda enak 0, saj smo za začetno stanje vzeli kar prvo meritev. Za tem residuali precej poskočijo, kar je bilo razvidno tudi iz slike (17). Residuali so nato pretežno enakomerno fluktuirajo okoli povprečja.
- Vsi elementi kovariančne matrike se hitro ustalijo na končni vrednosti, pojavijo pa se trije poskoki, ki sovpadajo

z zanko in obema ostrima ovinkoma, ki sledita za tem, kar je razumljivo, saj so to vse predeli, kjer se smer hitrosti hitro spreminja in jih zato dinamika sistema slabše opiše.

Ker je iz slike poti, težko oceniti, koliko boljša je naša napoved glede na uporabo surovih podatkov, si lahko ogledamo kako so videti residuali posamezne opazljivke.

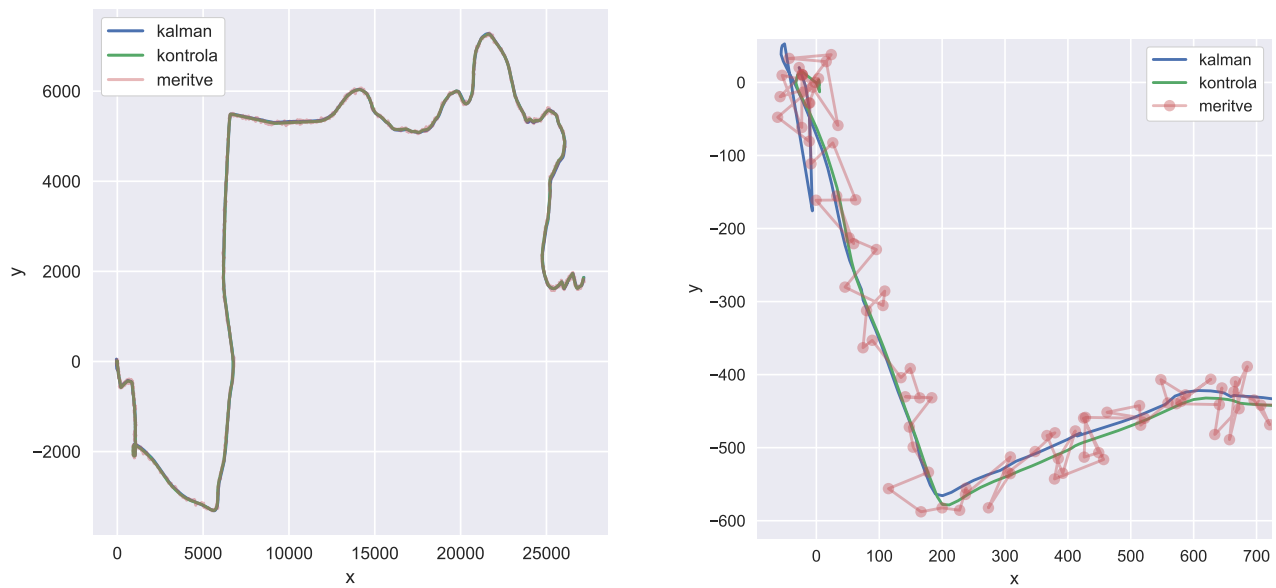


Slika 4: Primerjava med filtriranimi in izmerjenimi vrednostmi.

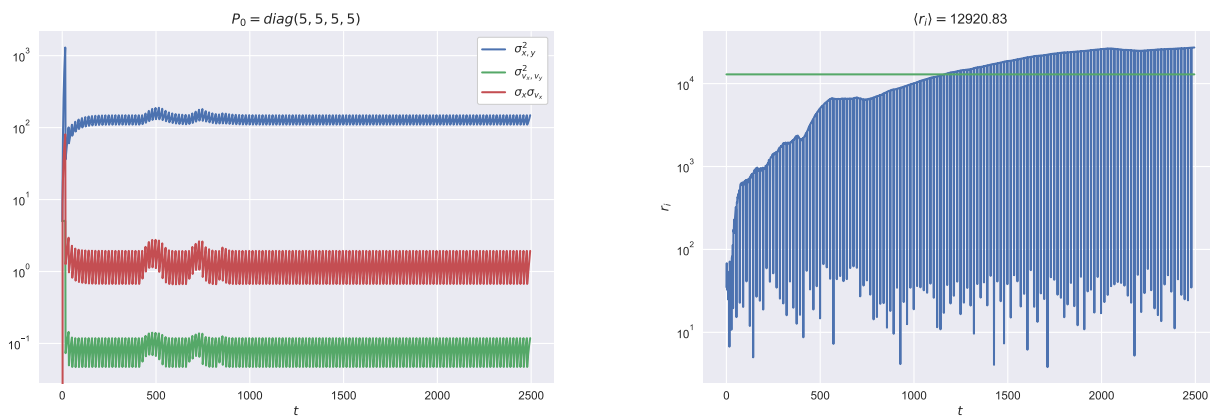
- Napoved lokacije smo s pomočjo Kalmanovega filtra uspeli bistveno izboljšati, kar se pozna tudi na raztrosu točk, ki so sedaj veliko bližje sredini, kar pomeni, da so bližje kontrolnim vrednostim.
- Napovedi hitrosti so zanimivo slabše glede na gole podatke, česar ne znam povsem pojasniti. Verjetno pa je del problema to, da je privzeta dinamika modela linearna, kar se izkaže za preslab približek resničnemu dogajanju.
- Z grafa napovedi hitrosti se lepo vidi tudi kako enakomerno, na oko Gaussovo, so merjene hitrosti raztrošene okoli pravih vrednosti, kar potrди, da je merilni šum res Gaussovski.

2.3 Vzorčenje vsake pete meritve lokacije in desete hitrosti

Poglejmo si, kako se rezultati filtriranja spremenijo, če namesto vseh meritev upoštevamo le vsako peto meritev položaja in vsako deseto hitrosti. Zaplet implementiramo v naš filter na način, da na vseh korakih, ko meritev ne upoštevamo, x_{n+1}^+ izračunamo zgolj na podlagi dinamike in podatkov o pospešku. Tako je H kar prazna matrika, vsako peto meritev spremenimo prva dva diagonalna člena v enici, vsako deseto meritev pa v identiteto.

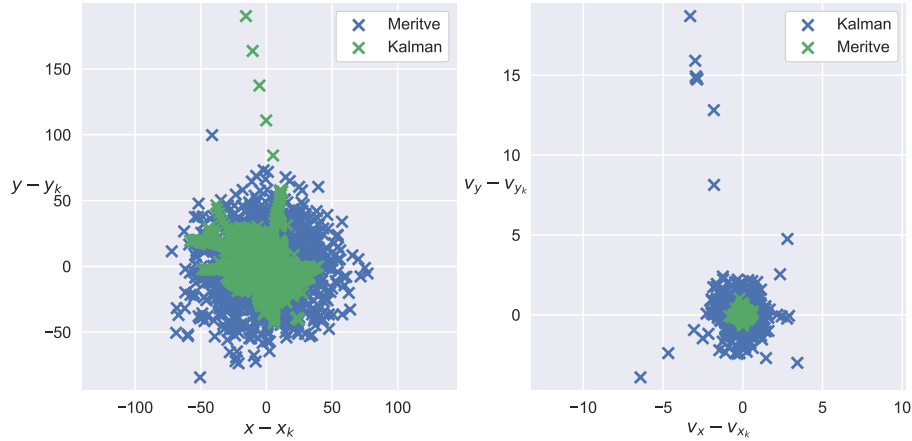


Slika 5: Napovedana pot pri uporabi le dela podatkov. Na desni je približan začetni del poti.



Slika 6: Elementi kovariančne matrike in residuali $\|z_{n+1} - H_{n+1}x_{n+1}^-\|$.

Primerjava med meritvami in filtriranimi podatki



Slika 7: Primerjava med filtriranimi in izmerjenimi vrednostmi.

- Z grafa poti je videti, da še vedno precej solidno sledimo dejanskim kontrolnim vrednostim, ima pa filter težave pri prvih nekaj točkah, kjer so že same meritve precej nekoherentne. Tam očitno dobi v sistem meritev, ki ga zavede in zato precej slabo oceni prvih nekaj metrov poti.
- Elementi kovariančne matrice se podobno kot prej razmeroma hitro ustalijo in doživijo enake poskoke ob težavnih delih poteh, a vseskozi oscilirajo, saj se variance ob korakih, ko ne prejmemo novih meritev povečujejo, nato pa znatno padejo, ko v sistem podamo podatek o meritvi.
- Podoben efekt opazimo tudi pri residualih, ki pravtako padejo, ko so nam na voljo meritve. Zanimivo je, da so odstopanja po poti vse večja, kar je morda posledica tega, da je zadnji del poti morda nekoliko zahtevnejši za opisati z manj točkami meritev, saj je bolj vijugast. Vseeno pa je videti kakor, da bi se napake zaradi pomanjkanja točk nekako seštevale.
- Točke so bistveno bolj raztrošene po grafu, še vedno pa opazimo enako efekt izboljšanja napovedi položaja, ne pa hitrosti.

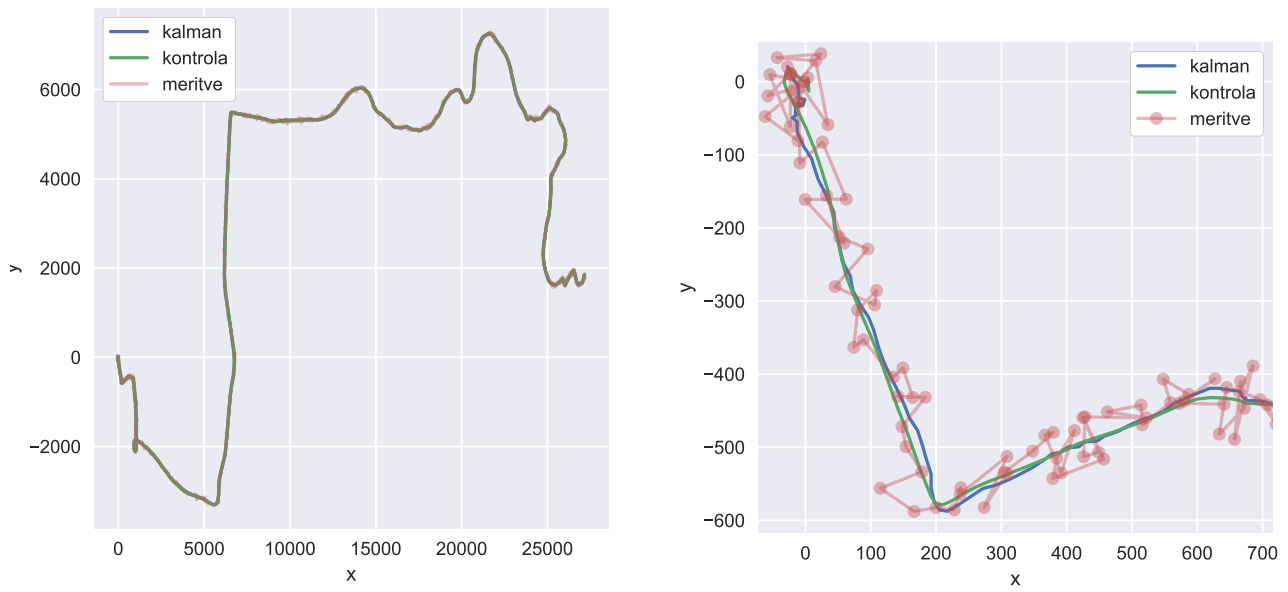
3 Rekonstrukcija poti s pomanjkljivimi podatki

3.1 Naloga

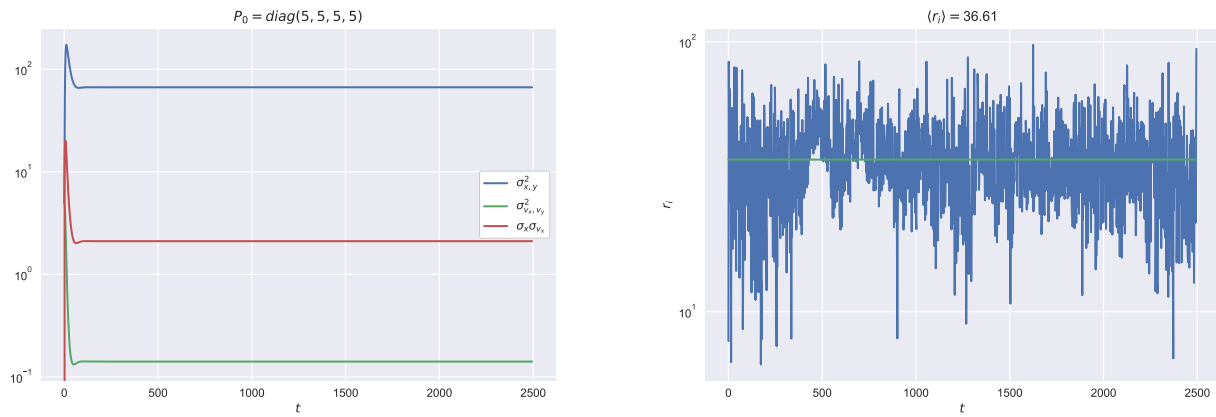
S telefonom lahko merimo le lokacijo in pospeške, informacij o hitrostih pa nimamo. Kako dobro lahko določite trajektorijo vožnje in potek hitrosti, če podatki o hitrosti niso na voljo? Kaj pa obratno, če so znane samo hitrosti in pospeški?

3.2 Rekonstrukcija poti brez podatkov o hitrostih

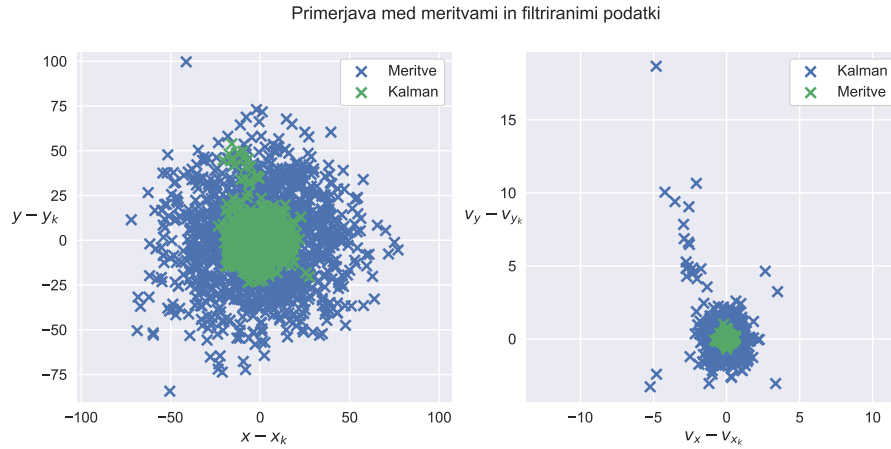
Poglejmo si, kako uspešno lahko sledimo poti, v kolikor se odpovemo podatkom o hitrosti - v matriki H zadnja dva diagonalna člena postavimo na 0.



Slika 8: Napovedana pot pri uporabi zgolj podatkov o položaju. Na desni je približan začetni del poti.



Slika 9: Elementi kovariančne matrike in residuali $\|z_{n+1} - H_{n+1}x_{n+1}^-\|$.

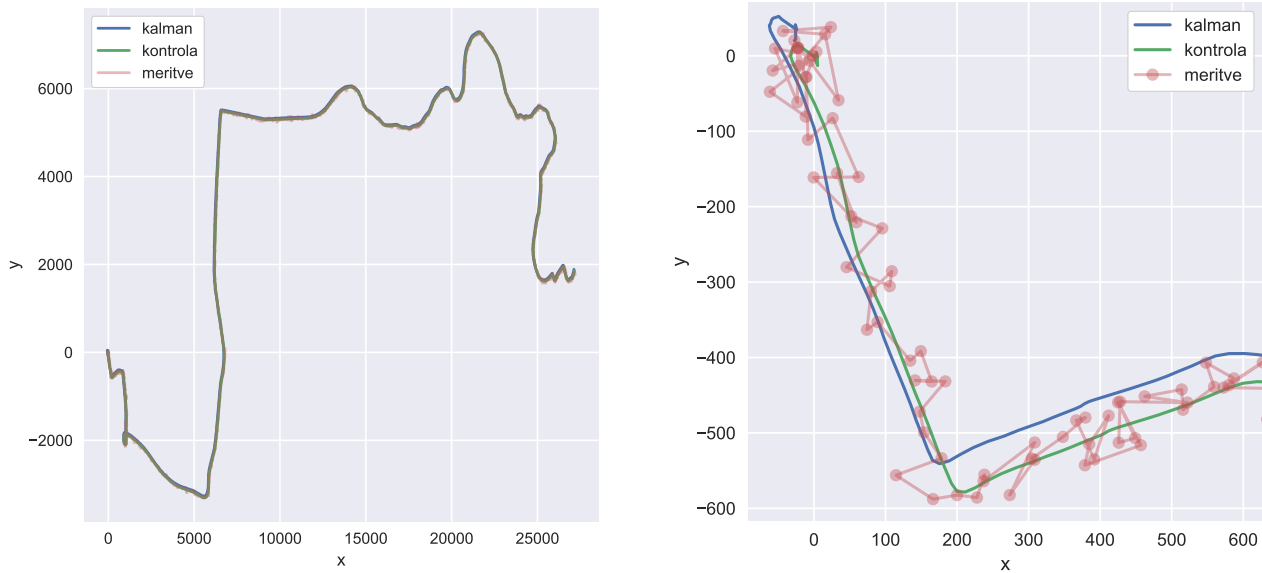


Slika 10: Primerjava med filtriranimi in izmerjenimi vrednostmi.

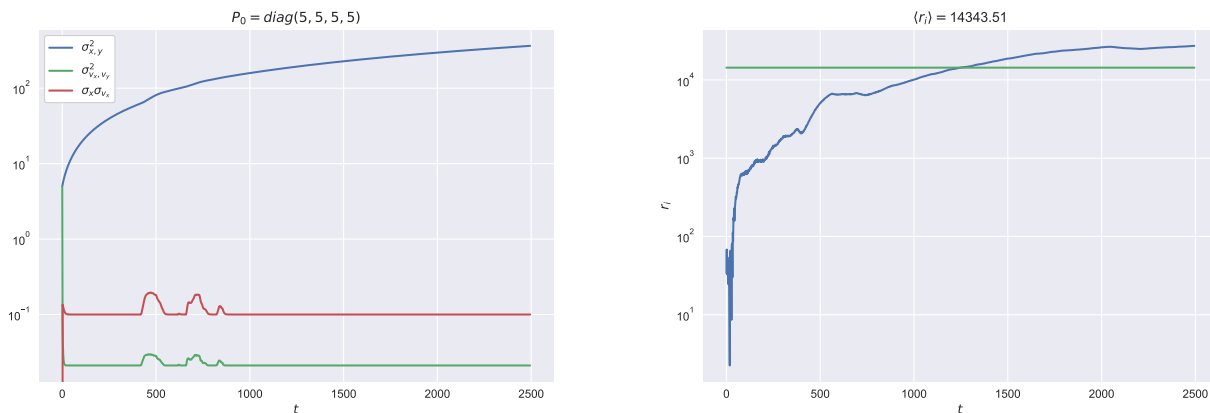
- Napovedana pot je videti celo nekoliko bolj pravilna, kot tista, ki smo jo dobili ob upoštevanju vseh meritev.
- Elementi kovariančne matrike sedaj prav nič ne poskočijo ob težavnejših delih, kar pomeni, da je model bolj prepričan v to, da je dobro 'zvožil' težje zavoje.
- Kljub boljšemu ujemanju položaja s kontrolo, je povprečen residual vseeno višji, kot je bil pri uporabi vseh podatkov, kar pomeni, da smo kontrolnim hitrostim v tem primeru slabše sledili. To pokaže tudi slika (10)

3.3 Rekonstrukcija poti brez podatkov položaju

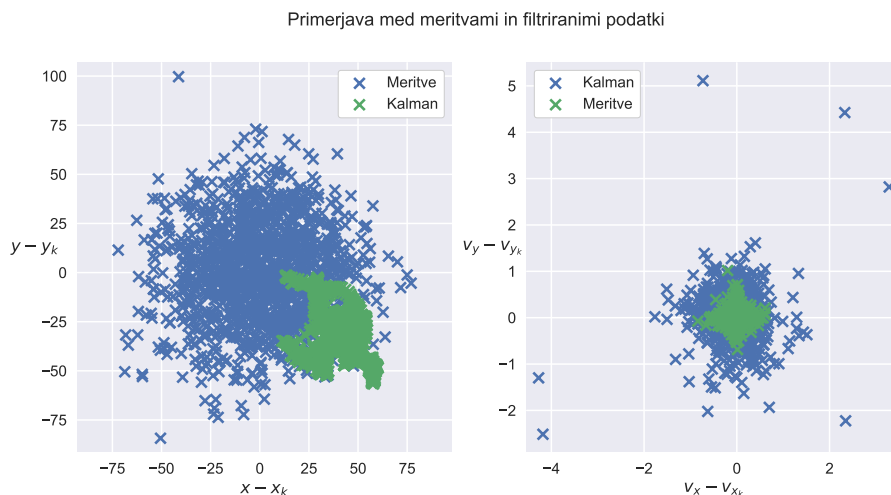
Sedaj obravnavajmo še obraten primer od prejšnjega, ko so nam na voljo zgolj podatki o hitrostih, o lokaciji pa ne.



Slika 11: Napovedana pot pri uporabi zgolj meritev hitrosti. Na desni je približen začetni del poti.



Slika 12: Elementi kovariančne matrike in residuali $\|z_{n+1} - H_{n+1}x_{n+1}^-\|$.

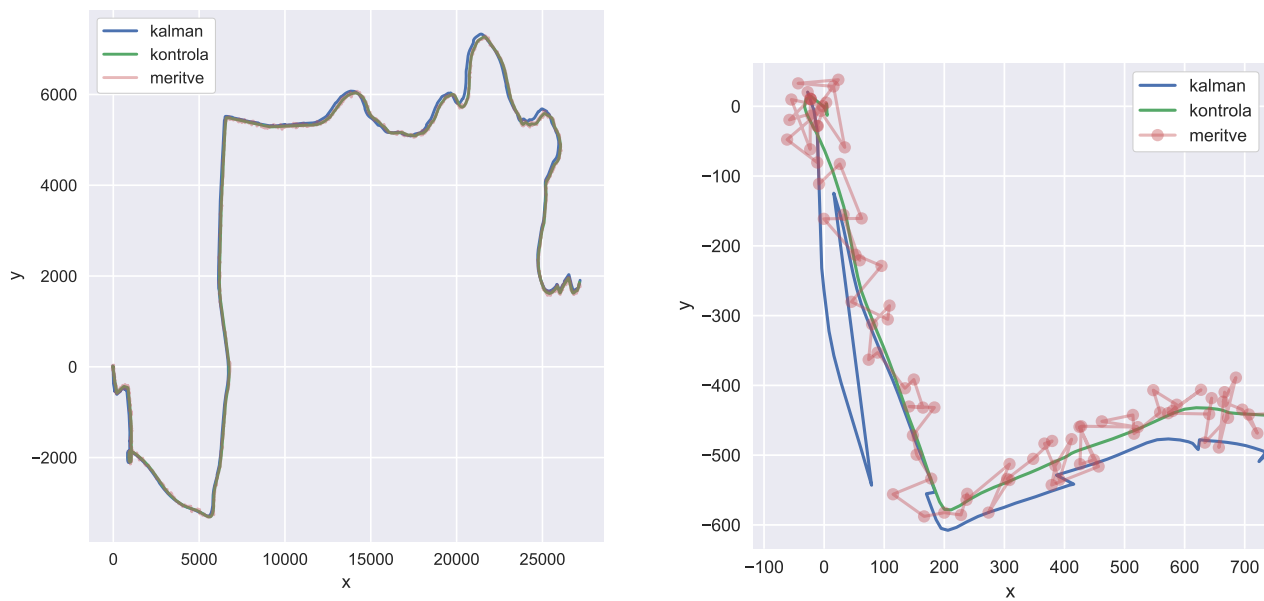


Slika 13: Primerjava med filtriranimi in izmerjenimi vrednostmi.

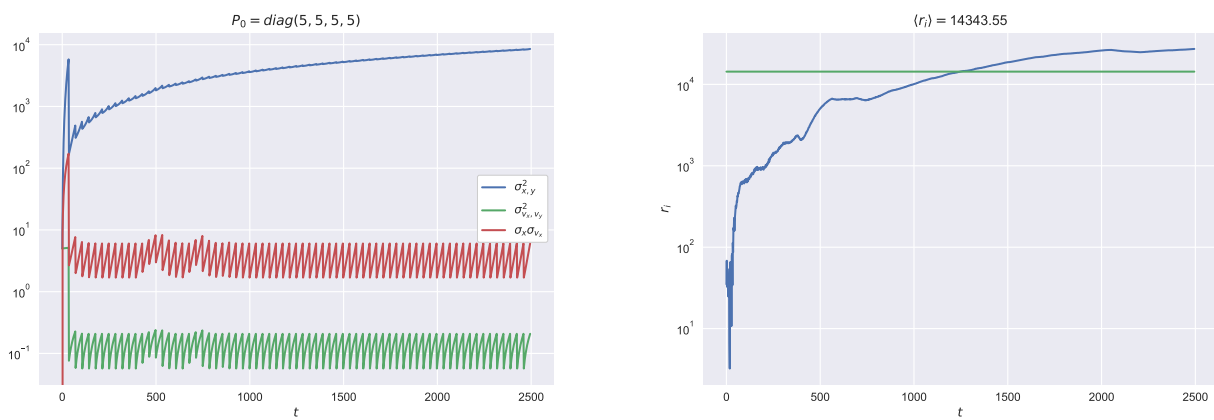
- Približna slika nam razkrije, da je ujemanje s kontrolno potjo v tem primeru razmeroma slabo, vidimo, da se na ravnih delih lahko pojavi nekakšen konstantni zamik od pravilne poti. Na teh delih se model ne uspe popraviti, saj ima dostop le do meritev o hitrostih, iz katerih ni možno zaznati, da smo pravzaprav nekoliko zgrešili pravo pot.
- Zaradi efekta zamika lege zato tudi varianca lege vseskozi raste. Ta informacija nam da celotno sliko - filter 'se zaveda', da ni siguren, če je na pravi lokaciji, ampak se ne zna popraviti na pravo mesto.
- Grafa raztrosa točk sta koherentna z vso dosedanja razlago - točke lege so si zelo skupaj, a so vse zamaknjen od sredine - filter skozi vsak korak prideluje večje napake. Hitrosti so vsaj na oko nekoliko bolj stisnjene v izhodišče, kar je pravitako smiselno.

3.4 Rekonstrukcija poti brez podatkov položaju in redkejšim vzorčenjem

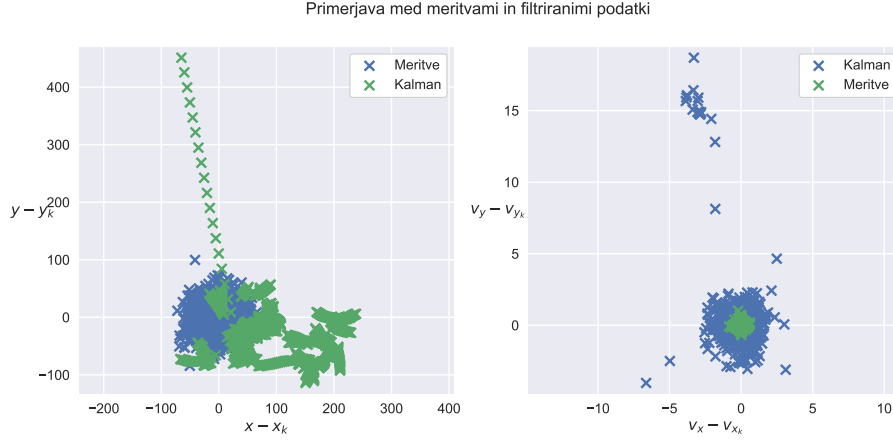
Obdržimo popolnoma enak model kot v prejšnjem razdelku, le da mu podajamo podatke o meritvah hitrosti le vsak peti korak.



Slika 14: Napovedana pot pri uporabi le dela podatkov o hitrosti in brez informacij legi. Na desni je približan začetni del poti.



Slika 15: Elementi kovariančne matrike in residuali $\|z_{n+1} - H_{n+1}x_{n+1}^-\|$.



Slika 16: Primerjava med filtriranimi in izmerjenimi vrednostmi.

Vse slike prikazujejo efekte opisane v prejšnjih razdelkih, le da so tu še močnejši - pot je približno prava, a zamaknjena, varianca lege konstantno narašča, ostali dve oscilirata ob vsaki novi podani meritvi, residuali konstantno naraščajo, točke z grafa primerjave so zelo raztresene, morda celo slabše kot gole meritve.

4 Realistična obravnava akcelerometra

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

$$\begin{aligned} u_n &= (0, 0, a_t \Delta t, a_r \Delta t), \\ B_n &= \begin{bmatrix} I_{2 \times 2} & 0 \\ 0 & B_n^{vv} \end{bmatrix}, \\ B_n^{vv} &= \frac{1}{\|v_n\|} \begin{bmatrix} v_x & -v_y \\ v_y & v_x \end{bmatrix} \end{aligned} \quad (4)$$

Nekoliko bolj zapletena je potem kovarianca meritve pospeška, saj moramo upoštevati še negotovost trenutne ocene hitrosti:

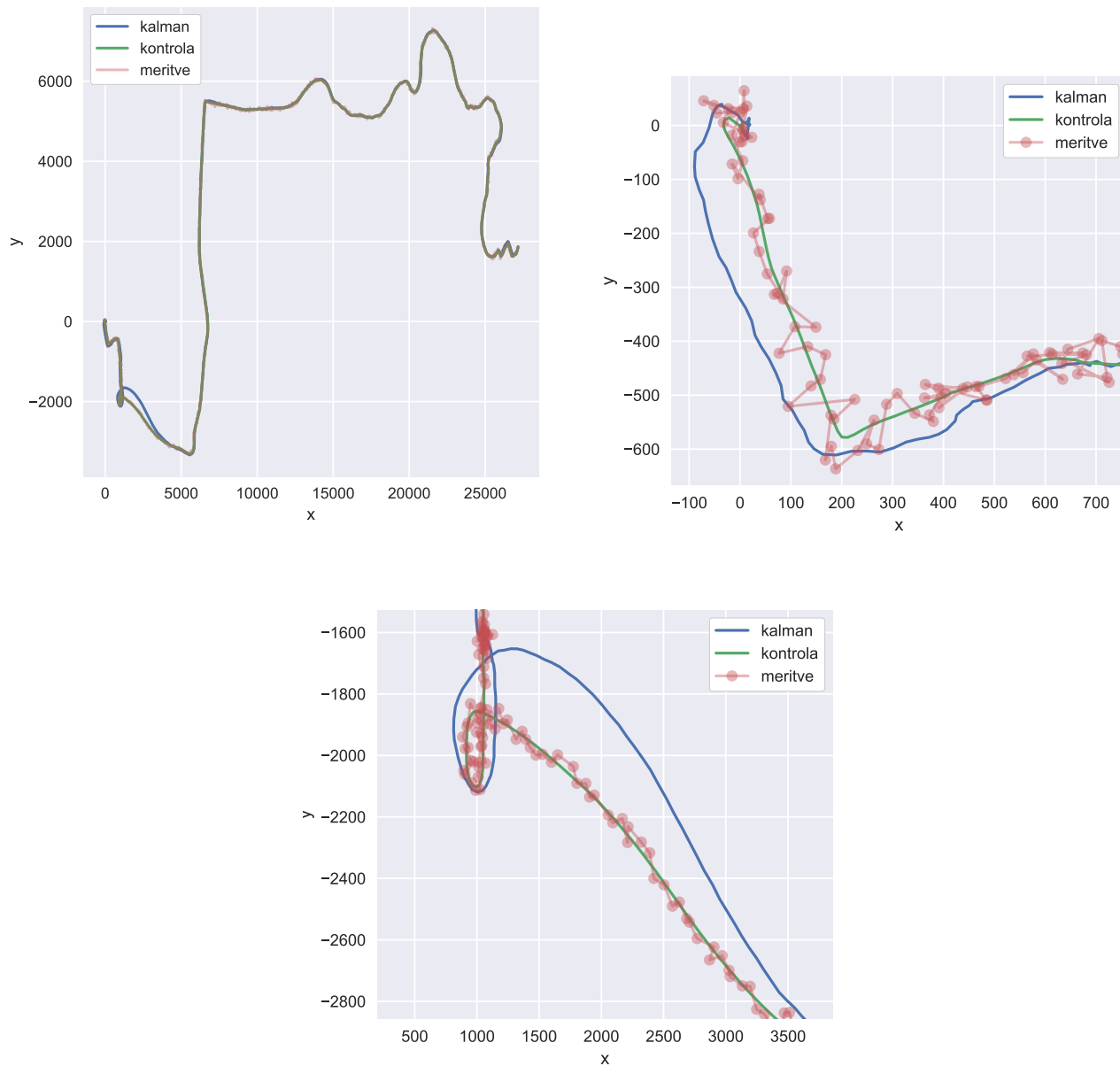
$$\begin{aligned} Q_n &= Q_{\text{od prej}} + \begin{bmatrix} 0 & 0 \\ 0 & Q_n^{vv} \Delta t \end{bmatrix}, \\ Q_n^{vv} &= \frac{R(\frac{\pi}{2}) v_n P_n^{vv} R(\frac{\pi}{2}) v_n}{\|v_n\|^4} \left[\left(B_n^{vv} R(\frac{\pi}{2}) a_n \right) \otimes \left(B_n^{vv} R(\frac{\pi}{2}) a_n \right) \right] \end{aligned} \quad (5)$$

Pri tem sta Q_{vv} in P_{vv} hitrostno-hitrostna bloka 4×4 kovariančnih matrik. Vektorja a_n in v_n sta pospešek in hitrost, ki ju z rotacijsko matriko $R(\frac{\pi}{2})$ zavrtimo v pozitivni smeri za $\frac{\pi}{2}$. Prvi člen je identičen kot prej, drugi člen pa upošteva, da projekcija negotovosti hitrosti prečno na smer gibanja pomeni nedoločenost kota in kviri zanesljivost pospeška. Če vam ta člen dela težave, ga lahko izpustite. Rekonstruiraj pot iz podatkov, danih v datoteki **kalman_relative_data.dat**, katere stolpci so čas t , GPS položaji (x, y) ter pospeška (a_t, a_r) . Spet lahko izbiraš, kako pogosto so na voljo meritve položaja.

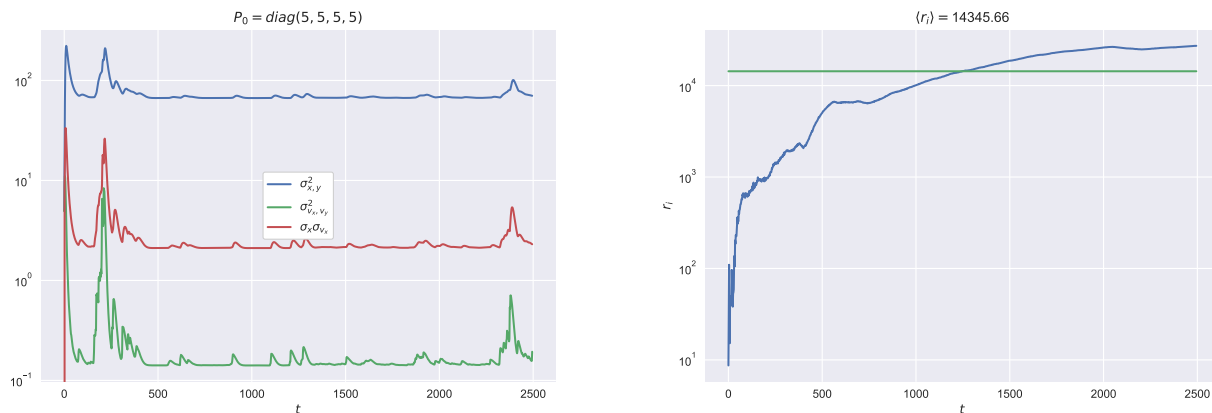
Pogledali si bomo dva primera vzorčenja; takega ko so nam na voljo vsi podatki položaja in pa takšnega, ko nekje sredi poti izgubimo GPS signal in nam podatki o legi niso več na voljo. V obeh primerih ne bomo imeli na voljo podatkov o hitrostih.

4.1 Konstantno vzorčenje položaja

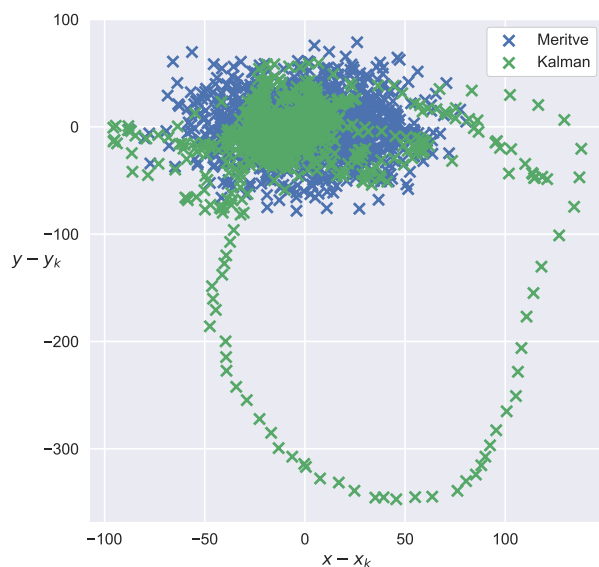
V modelu nimamo dostopa do meritev hitrosti, a bomo za začetno stanje x_0^- vseeno uporabili kar hitrosti iz kontrolne datoteke.



Slika 17: Napovedana pot - na desni je približan začetni del poti, spodaj pa težavni ovinek.



Slika 18: Elementi kovariančne matrike in residuali $\|z_{n+1} - H_{n+1}x_{n+1}^-\|$.

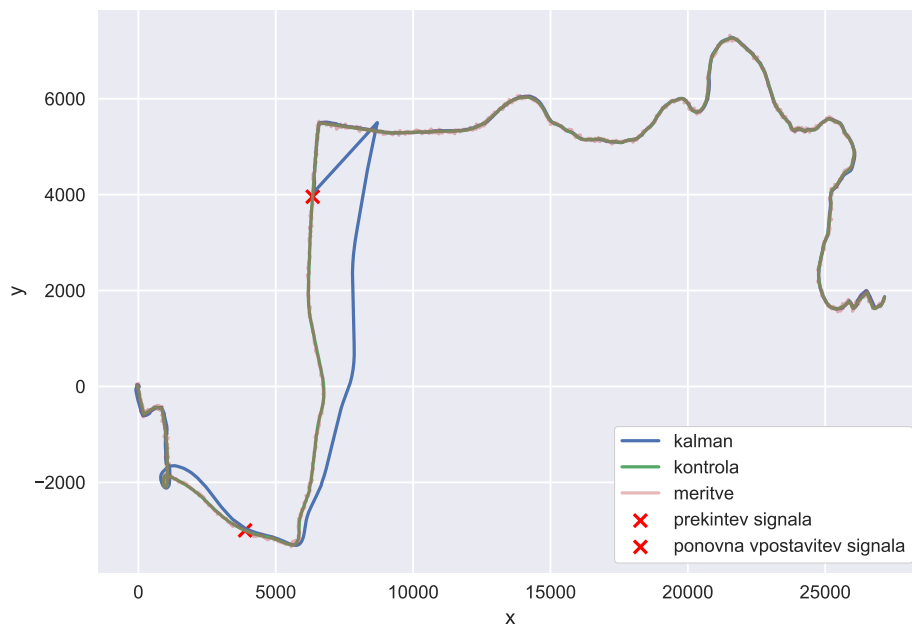


Slika 19: Primerjava med filtriranimi in izmerjenimi vrednostmi.

- Filtriranje je bilo v tem primeru manj uspešno, saj se napovedana pot precej slabo ujema s kontrolno. Najbolj težaven del je zagotovo zanka, kjer so veliki radialni pospeški, kar sistem očitno zmede.
- Z grafa spreminjanja vrednosti kovariančne matrike vidimo, da je sistem precej negotov ob mnogih ovinkih.
- Z grafa odstopanja lege lahko zelo jasno vidimo katere točke smo pridelali ob tisti zanki, na splošno pa so točke kar močno raztresene, vseeno pa nekoliko manj od samih meritev.

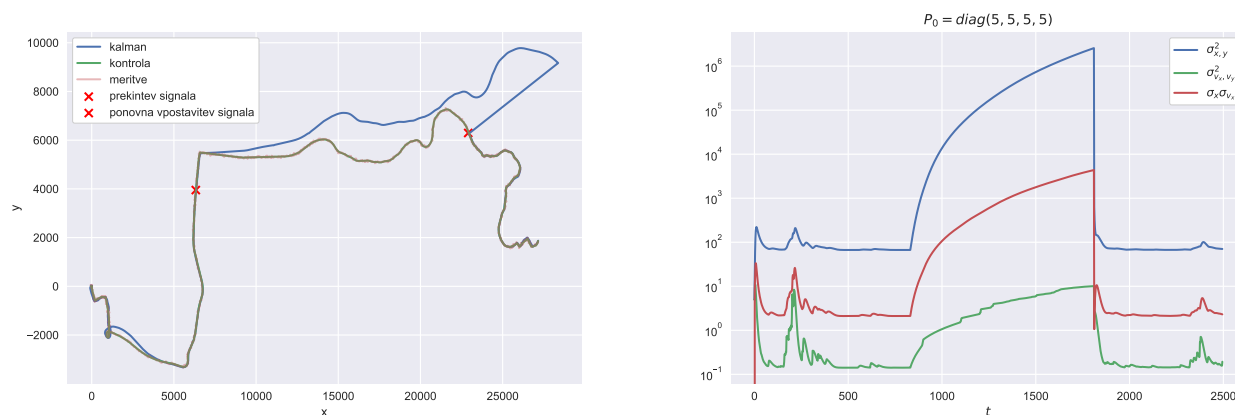
4.2 Izpad GPS signala

Zelo življenjski primer predstavlja situacija, ko sredi poti na telefonu izgubimo GPS signal in so nam tako na voljo le še podatki iz akcelerometra. Prikazani sta dve simulaciji, pri katerih smo naključno žrebali kdaj izgubimo in pridobimo signal nazaj.



Slika 20: Napoved poti ob izgubi signala tik pred prvim ostrim ovinkom.

S takšnim filtrom pričakovano spet precej slabo zvozimo zanko pri vključevanju na avtocesto, je pa zanimivo, kako kljub brez podatkov o legi pretežno dobro zvozimo naslednji ovinek. Lahko sklepamo, da model nekoliko podcenjuje kako veliki morajo biti radialni pospeški, saj ob obeh ovinkih nekoliko premalo zavije.



Slika 21: Napoved poti ob izgubi signala tik pred drugim ostrim ovinkom in graf spreminjanja elementov kovariancijske matrike.

Tudi v tem primeru lahko opazimo podobno obnašanje ob prejšnjem, pravtako je zanimivo, kako hitro se sistem popravi, ko mu spet podamo podatke o legi, praktično v naslednjem koraku je spet na pravi poti, kar res pokaže, za kako močan algoritem gre - dobro se 'zaveda', da najverjetneje ni na pravi poti in se zelo hitro popravi, ko dobi zadostno informacijo. Vse to je vidno tudi iz spreminjanja varianc kovariancijske matrike.