SMC-menetelmät

Laskennallinen tilastotiede - Harjoitustyö

Lasse Rintakumpu

29 April, 2021

Sisällys

1 Johdanto **1**
1.1 Suodinongelma . . . 2
1.2 Historiaa . . . 2

2 Bayesilainen suodin **3**

3 Hiukassuodin-algoritmi **3**
3.1 Parametrien valinta . . . 4
3.1.1 Otoskoon N valinta . . . 4
3.1.2 Uudelleenotantamenetelmän valinta . . . 5
3.1.3 Ehdotusjakauman valinta . . . 5
3.2 Konvergenssituloksia . . . 6
3.3 Marginaalijakauma . . . 6
3.4 Aikakompleksisuus . . . 7

4 Paikannusesimerkki **7**
4.1 Datan kuvaus . . . 7
4.2 Koeasetelma . . . 7
4.3 Malli . . . 7
4.4 Algoritmi . . . 7
4.5 Tulokset . . . 7

5 Lopuksi **8**

6 Lähteet **8**

1 Johdanto

SMC-menetelmät (Sequential Monte Carlo -mentelmät, tunnetaan myös nimellä hiukkassuotimet) ovat joukko 90-luvulta eteenpäin kehitettyjä Monte Carlo -algoritmeja, joiden avulla voidaan ratkaista ns. suodinongelma, kun ongelma on epälineaarinen ja/tai ongelmaan liittyvä kohina ei noudata normaalijakaumaa. Tämän tutkielman tavoitteena on esittää pääpiirteittäin SMC-menetelmien teoria sekä joitakin menetelmäperheeseen kuuluvia algoritmeja. Tutkielman esitykset seuraavat erityisesti Simo Särkän kirjaa *Bayesian Filtering and Smoothing* (2013) sekä Fredrik Gustafssonin artikkelia "Particle Filter Theory and Practice with Positioning Applications" (2010). SMC-menetelmille on lukuisia sovellutuksia esimerkiksi Bayesilaisessa tilastotieteessä, fysiikassa ja robotiikassa. Tämän tutkielman lopussa tarkastellaan hiukkassuotimen käyttöä paikannussovelluksessa.

1.1 Suodinongelma

Stokastisten prosessien teoriassa suodinongelmaksi kutsutaan tilannetta, jossa halutaan muodostaa paras mahdollinen estimaatti jonkin järjestelmän tilan arvoille, kun ainoastaan osa tiloista voidaan havaita ja/tai havaintoihin liittyy kohinaa. Tavoitteena on siis laskea jonkin Markov-prosessin posteriorijakauma kyseisten havaintojen perusteella. Tässä tutkielmassa keskitytään erityisesti epälineaarisen ns. Markovin piilomallin posteriorijakauman Bayesilaiseen ratkaisuun. Ongelmaa havainnollistaa kaavio (1).

$$\begin{array}{ccccccc} X_1 & \longrightarrow & X_2 & \longrightarrow & X_3 & \longrightarrow & \dots \quad \text{piilossa olevat tilat} \\ \downarrow & & \downarrow & & \downarrow & & \\ Y_1 & & Y_2 & & Y_3 & \dots & \text{havainnot} \end{array} \tag{1}$$

Ongelmassa tiedämme, miten havaitut muuttujat kytkeytyvät "piilossa oleviin" muuttujiin sekä osaamme sanoa jotain tilamuuttujien todennäköisyyksistä. Oletamme lisäksi, että piilossa olevat tilat muodostavat Markovin ketjun. Kun aika-avaruus on diskreetti ja merkitsemme piilossa olevan prosessin tilaa ajanhetkellä k x_k ja havaittua prosessia y_k, meillä on siis olemassa mallit

$$x_{k+1} = f(x_k, \nu_k) \tag{2}$$
$$y_k = h(x_k) + e_k. \tag{3}$$

Lisäksi tiedetään prosessin alkuhetken jakauma $x_1 \sim p_{x_1}$, tähän liittyvän kohinprosessin jakauma $\nu_k \sim p_{\nu_k}$ sekä malliin y_k liittyvä kohina $e_k \sim p_{e_k}$. Mallit voidaan yleisemmällä tasolla esittää myös muodossa

$$x_{k+1} \sim p(x_{k+1}|x_k) \tag{4}$$
$$y_k \sim p(y_k|x_k). \tag{5}$$

Tutkielman teoriaosassa käytetään ensisijaisesti yhtälöiden (4) ja (5) muotoilua. Empiirisessä osassa palataan yhtälöiden (2) ja (3) muotoiluun.

Näin määritettyjen mallien avulla SMC-menetelmät estimoivat sekventiaalisesti tilojen X_k arvot minä hyvänsä ajan hetkellä k, kun ainoastaan prosessi $Y_1, \dots, Y_k$ tunnetaan. Estimaatit saadaan posteriorijakaumasta $p(x_k|y_1, y_2, \dots, y_k)$, jonka approksimaatio muodostetaan Bayesilaisittain havaintojen pohjalta. Kuten mainittua, ei SMC-perheen algoritmeja käytettäessä mallin $y_k = h(x_k) + e_k$ tarvitse olla lineaarinen eikä kohinaprosessien noudataa normaalijakaumaa. SMC-menetelmissä stokastisen prosessin posteriorijakauman esittämiseen käytettyjä otoksia kutsutaan myös partikkeleiksi. Suodinongelmaa lähellä on myös ns. tasoitusongelma (smoothing problem), jossa ollaan kiinnostuneita prosessin x_k posteriorijakaumasta $p(x_k|y_k)$ jokaisena ajanhetkenä $1, \dots, k$ ei ainoastaan haluttuna ajanhetkenä k. Tämä tutkielma keskittyy yksin suodinongelman ratkaisemiseen, mutta SMC-algoritmin marginalisointia koskevassa luvussa viitataan myös tasoitusongelman ratkaisuun.

1.2 Historiaa

Lineaarisen suodinongelman ratkaisu on tunnettu ... Epälineaariselle ei-Gaussilaiselle mallille ... Muita vaihtoehtoja ovat EKF, UKF, QKF.

It should b e stressed that both EKF and UKF approximate the model and propagate Gaussian distributions representitive of the post erior while the PMF uses the original model and approxi mates the posterior over a grid. The particle filt er (PF) also provides a numerical approximati on to the nonlinear filt eri ng prob lem similar to the PMF but uses an ad aptive stochastic grid that aut omatically select s relevant grid points i n the st ate space, and in contrast to the PMF, the st and ard PF has linear complexity i n the number o f grid points. The first traces of the PF date b ack to the 1950s [11 , 12] , and the control community made some attempts in the 1970s [13 , 14]. However, the PF era st arted with the semi nal paper [15] , and the i ndependent development s in [16 , 17]. Here, an important res ampli ng step was i ntroduced.

The ti mi ng for proposing a general solution to the nonlinear filteri ng prob lem was perfect in that the computer development enabled the use of computationally complex algorithms in quite realistic prob lems. Si nce the paper [15] the research has steadily i ntensified; see the article collection [18] , the surveys [19-22] , and the monograph [23]. Fig. I illustrat es how the number of papers i ncreases exponenti ally each y ear, and the s ame appears to be true for applied papers. The PFs may be a serious alternative for real-ti me applications classically approached by the (E)KF. The more nonlinear model, or the more non-Gaussian noise, the more potential PFs have, especi ally in applications where computational power is rather cheap, and the s ampling rate i s moderate

Monte Carlo -ratkaisuja (esim. Princeton)

Ensimmäisen epälineaarisen suodinongelman Bayesilaisen/MC-ratkaisun esittivät Gordon, Salmond ja Smith artikkelissaan "Novel approach to nonlinear/non-Gaussian Bayesian state estimation" (1993). Gordonin, Salmondin ja Smithin ratkaisu eroaa notaatioltaan hieman tässä tutkielmassa esitetystä, mutta on olennaisesti sama. Suurin ero tämän tutkielman SMC-algoritmin sekä alkuperäisen SMC-algoritmin välillä on XXX MITEN XXX. Artikkelissa ratkaisu kulki nimellä "bootstrap filter", saapasremmisuodin. MIKSI Termiä hiukkassuodin käytti ensimmäisen kerran Del Moral artikkelissa "Nonlinear Filtering: Interacting Particle Resolution" (1996), SMC-menetelmät termiä Liu ja Chen artikkelissa "Sequential Monte Carlo Methods for Dynamic Systems" (1998). Tässä tutkielmassa pyritään korostamaan suotimien yhteyttä Monte Carlo -algoritmeihin ja käytetään siksi termiä SMC-menetelmät. Poikkeuksen tähän tekee varsinainen esitetty algoritmi, jota kutsutaan tutkielmassa hiukkassuodin-algoritmiksi.

2 Bayesilainen suodin

Ennen SMC-algoritmia käydään läpi algoritmeissa käytetty yleinen Bayesilainen posteriorijakauman laskenta. Esitys noudattaa Fredrik Gustafssonin artikkelia "Particle Filter Theory and Practice with Positioning Applications" (2010). Bayesilainen ratkaisu tilavektorin posteriorijakaumalle $p(x_k|y_{1:k})$ saadaan seuraavalla rekursiolla (käydään läpi jokaiselle ajanhetkelle $k = 1, \ldots, t$). Lasketaan ensin

$$p(x_k|y_{1:k}) = \frac{p(y_k|x_k)p(x_k|y_{1:k-1})}{p(y_k|y_{1:k-1})}, \tag{6}$$

joka saadaan suoraan Bayesin kaavasta $P(A|B) = P(B|A)P(A)/P(B)$. Normalisointivakio lasketaan integraalina

$$p(y_k|y_{1:k-1}) = \int_{\mathbb{R}^{n_x}} p(y_k|x_k)p(x_k|y_{1:k-1})\, dx_k, \tag{7}$$

joka saadaan kokonaistodennäköisyyskaavasta $P(A) = \mathbb{E}[P(A|X)] = \int_{-\infty}^{\infty} P(A|X = x)f_X(x)\, dx$. Merkintä R^{n_x} vastaa tässä piilossa olevien muuttujien dimensiota n.

Lopuksi lasketaan päivitysaskel ajalle, joka saadaan edelleen kokonaistodennäköisyydellä

$$p(x_{k+1}|y_{1:k}) = \int_{\mathbb{R}^{n_x}} p(x_{k+1}|x_k)p(x_k|y_{1:k})\, dx_k\,. \tag{8}$$

Rekursion avulla voidaan laskea $p(x_k|y_{1:k})$ käymällä rekursio läpi k kertaa.

3 Hiukkassuodin-algoritmi

Alla esitetään hiukkassuodin-SMC-algoritmi Bayesilaisen, epälineaarisen suodinongelman ratkaisemiseksi. Tässä luvussa esitetään algoritmi, joka hyödyntää luvussa 2. kuvattua Bayesilaista suodinta epälineaarisen suodinongelman ratkaisemiseen.

Kun uudelleenotanta tehdään jokaisella algoritmin iteraatiolla, on kyseessä SIS-algoritmi. Kun uudelleenotanta tehdään esimerkiksi edellisessä alaluvussa kuvatun kynnysarvoehdon $\hat{N}_{eff} < N_{th}$ täyttessä, on kyseessä SIR-algoritmi. Algoritmin kuvauksessa käytetään notaatiota x_k^i, joka tarkoittaa, että tila x_k käy ajanhetkellä k gridin pisteessä x^i. Notaatiota tarvitaan, koska SMC-algoritmin läpikäymä gridi muuttuu ajan funktiona.

Ennen varsinaista algoritmia käsitellään algoritmiin liittyvän uudelleenotantamenetelmän, partikkelien määrän / otoskoon ja ehdotusjakauman valinta. Lopuksi esiteetään algoritmin konvergenssia, marginaalijakaumaa sekä aikakompleksisuutta koskevia tuloksia. Esitetty algoritmi perustuu Fredrik Gustafssonin artikkeliin "Particle Filter Theory and Practice with Positioning Applications" (2010).

Algorithm 1: Hiukassuodin

Data: Generoidaan $x_1^i \sim p_{x_0}$ missä $i = 1, \ldots, N$ ja jokainen partikkeli saa saman painon $w_{1|0}^i = 1/N$.
Result: Posteriorijakauma $p(x_{1:k}|y_{1:k})$.
begin
  for $k = 1, 2, \ldots, t$ **do**
    for $i = 1, 2, \ldots, N$ **do**
      begin
        Päivitetään painot. $w_{k|k}^i = \frac{1}{c_k} w_{k|k-1}^i p(y_k|x_k^i)$,
        missä normalisointipaino on $c_k = \sum_{i=1}^N w_{k|k-1}^i p(y_k|x_k^i)$.
      begin
        Estimoidaan p.
        Lasketaan tiheydelle approksimaatio $\hat{p}(x_{1:k}|y_{1:k}) = \sum_{i=1}^N w_{k|k}^i \delta(x_{1:k} - x_{1:k}^i)$,
        missä $\delta(x)$ on Diracin deltafunktio.
      begin
        Lasketaan efektiivinen otoskoko $\hat{N}_{eff} = \frac{1}{\sum_i (w_{k|k}^i)^2}$.
      if $\hat{N}_{eff} < N_{th}$ **then**
        begin
          Otetaan uudet N otosta palauttaen joukosta $\{x_{1:k}^i\}_{i=1}^N$, missä otoksen i todennäköisyys on $w_{k|k}^i$.
      if $k < t$ **then**
        begin
          Aikapäivitys.
          Luodaan ennusteet partikkeleille ehdotusjakaumasta $x_{k+1}^i \sim q(x_{k+1}|x_k^i, y_{k+1})$,
          päivitetään partikkelien painot tärkeytysotannalla sen mukaan kuinka todennäköisiä partikkelien ennusteet ovat $w_{k+1|k}^i = w_{k|k}^i \frac{p(x_{k+1}^i|x_k^i)}{q(x_{k+1}^i|x_k^i, y_{k+1})}$.

3.1 Parametrien valinta

Ennen algoritmin suorittamista valitaan ehdotusjakauma $q(x_{k+1}|x_{1:k}, y_{k+1})$, uudelleenotantamenetelmä sekä partikkelien määrä N. Ehdotusjakauman ja uudelleenotantamenetelmän valinnassa tärkeimpänä päämääränä on välttää otosten ehtymistä, kun taas partikkelien määrä säätelee kompromissia algoritmin suorituskyvyn ja tarkkuuden välillä.

3.1.1 Otoskoon N valinta

Yleispätevää sääntöä otoskoon/partikkelien lukumäärän N valinnalle on vaikeaa antaa, sillä vaadittava estimointitarkkuus riippuu usein käsillä olevasta ongelmasta. Gordon &al. (1993) esittävät kuitenkin kolme tekijää, jotka vaikuttavat partikkelien lukumäärän valintaan

a. tila-avaruuden ulottuvuuksien lukumäärä n_x,
b. tyypillinen päällekäisyys priorin ja uskottavuuden välillä
c. sekä tarvittava aika-askeleiden lukumäärä.

Ensimmäisen tekijän vaikutus on selvä. Mitä useammassa ulottuvuudessa otantaa tarvitsee tehdä, sen korkeammaksi on N asetettava, jotta jokainen ulottuvuus pystytään kattamaan. Tekijät (b) ja (c) puolestaan seuraavat uudelleenotannasta. Jos se osa tila-avaruutta, jossa uskottavuus $p(y_k|x_k)$ saa merkittäviä arvoja on pieni verrattuna siihen osaan, jossa priorijakauma $p(x_k|y_{1:k-1})$ saa merkittäviä arvoja, suuri osa partikkeleista saa pieniä painoja eivätkä näin valikoidu uudelleenotantaan.

Yleisesti ottaen N kannattaa asettaa sellaiseksi, että se paitsi tuottaa riittävän tarkan estimaatin, on se käytettävissä olevan laskentatehon sekä vaadittavan laskentanopeuden kannalta järkevää. Tähän palataan tutkielman lopuksi empiirisessä paikannusesimerkissä.

3.1.2 Uudelleenotantamenetelmän valinta

Ilman uudelleenotantaa on todennäköistä, että algoritmi alkaa kärsiä otosten ehtymisestä. Toisin sanoen kaikki painot alkavat keskittyä vain muutamalle partikkelille. Uudelleenotanta tarjoaa osittaisen ratkaisun tähän ongelmaan, mutta hävittää samalla informaatiota ja siten lisää satunnaisotantaan liittyvää epävarmuutta. Yleisesti ottaen kannattaa uudelleenotanta aloittaa vasta siinä vaiheessa algoritmin suorittamista, kun siitä on otosten ehtymisen kannalta hyötyä.

Jos alla esitetyssä algoritmissa uudelleenotanta suoritetaan jokaisella algoritmin läpikäynnillä on kyseessä ns. SIR-algoritmi (sequential importance resampling). Vaihtoehtona on hyödyntää tärkeytysotantaa ja suorittaa uudelleenotanta ainoastaan, kun otoskoon ehtymisen mittarina käytettävä efektiivinen otoskoko painuu jonkin kynnysarvon alapuolelle. Tätä kutsutaan SIS-algoritmiksi (sequential importance sampling).

Efektiivinen otoskoko saadaan laskettua variaatiokertoimesta c_ν kaavalla

$$N_{eff} = \frac{N}{1 + c_\nu^2(w_{k|k}^i)} = \frac{N}{1 + \frac{\mathrm{Var}(w_{k|k}^i)}{(\mathbb{E}[w_{k|k}^i])^2}} = \frac{N}{1 + N^2\mathrm{Var}(w_{k|k}^i)}. \tag{9}$$

Näin laskettu efektiivinen otoskoko maksimoituu ($N_{eff} = N$), kun kaikille painoille pätee $w_{k|k}^i = 1/N$ ja minimoituu ($N_{eff} = 1$), kun $w_{k|k}^i = 1$ todennäköisyydellä $1/N$ ja $w_{k|k}^i = 0$ todennäköisyydellä $(N-1)/N$. Tästä saadaan effektiiviselle otoskoolle laskennallinen approksimaatio

$$\hat{N}_{eff} = \frac{1}{\sum_i (w_{k|k}^i)^2}. \tag{10}$$

Sekä määritelmälle (9) että (10) pätee $1 \leq \hat{N}_{eff} \leq N$. Yläraja saavutetaan, kun jokaisen partikkelin paino on sama. Alarajalle puolestaan päädytään, kun kaikki paino päätyy yksittäiselle partikkelille. Tästä saadaan määriteltyä algoritmille SIS-uudelleenotantaehto $\hat{N}_{eff} < N_{th}$. Gustafsson (2010) esittää uudelleenotannan kynnysarvoksi esimerkiksi $\hat{N}_{th} = 2N/3$.

Uudelleenotanta ei muuta approksimoitavan jakauma p odotusarvoa, mutta se lisää jakauman Monte Carlo -varianssia. On kuitenkin olemassa uudelleenotantamenetelmiä, jotka pyrkivät minimoimaan tämän varianssin lisäyksen. Varianssitarkastelu jätetään tämän tutkielman ulkopuolelle.

3.1.3 Ehdotusjakauman valinta

Yksinkertaisin valinta ehdotusjakaumalle on $q(x_{1:k}|y_{1:k})$ eli toisin sanoen jokaisella algoritmin suorituskerralla käydään läpi koko polku $1:k$. Tämä ei kuitenkaan ole tarkoituksenmukaista, erityisesti jos kyseessä on reaaliaikainen sovellutus. Kirjoitetaan tämä ehdotusjakauma nyt muodossa

$$q(x_{1:k}|y_{1:k}) = q(x_k|x_{1:k-1}, y_{1:k})q(x_{1:k-1}|y_{1:k}). \tag{11}$$

Jos yhtälöstä (11) poimitaan ehdotusjakaumaksi ainoastaan termi $q(x_k|x_{1:k-1}, y_{1:k})$ saadaan tämän avulla muodostettua hyvä approksimaatio arvoille x_k. Tämä on suodinongelman kannalta riittävää, koska olemme kiinnostuneita ainoastaan posteriorijakaumasta ajanhetkellä k (tasoitusongelmassa tarvitsisimme koko polun $x_{1:k}$). Alla tarkastellaan edelleen Gustafssonia (2010) seuraten kahta ehdotusjakauman valintatapaa, prioriotantaa (prior sampling) sekä uskottavuusotantaa (likelihood sampling).

Ennen otannan tarkastelua määritellään mallille signaali-kohinasuhde uskottavuuden maksimin ja priorin maksimin välisenä suhteena

$$\text{SNR} \propto \frac{\max_{x_k} p(y_k|x_k)}{\max_{x_k} p(x_k|x_{k-1})}. \tag{12}$$

Kirjoitetaan lisäksi yhtälöt MUODOSSA

Kun suhde (12) on matala, on prioriotanta luonnollinen valinta. Tässä käytetään ehdotusjakauman tilavektorin ehdollista prioria eli

$$q(x_k|x_{1:k-1}, y_k) = p(x_k|x^i_{k-1}). \tag{13}$$

Kun taas signaali-kohinasuhde on kohtalainen tai korkea, on parempi käyttää ehdotusjakaumana skaalattua uskottavuusfunktiota

$$q(x_k|x_{1:k-1}, y_k) \propto p(y_k|x_k). \tag{14}$$

Tässä tapauksessa vaaditaan lisäksi, että uskottavuusfunktio on integroituva.

3.2 Konvergenssituloksia

Alla esitetään kaksi algoritmiin liittyvää konvergenssitulosta, se kuinka hyvin esitetyllä SMC-algoritmilla arvioitu posterioritiheys $\hat{p}(x_{1:k}|y_{1:k})$ approksimoi todellista tiheysfunktiota $p(x_{1:k}|y_{1:k})$ sekä mikä on approksimaation keskineliövirhe. Tulokset noudattavat Crisanin ja Doucet'n artikkelia "Convergence of Sequential Monte Carlo Methods" (2000).

Konvergenssitulos 1: Kun $N \to \infty$ algoritmille pätee $\forall k$ tulos $\hat{p}(x_{1:k}|y_{1:k}) \xrightarrow{a.s.} p(x_{1:k}|y_{1:k})$.

Konvergenssitulos 2: MSE-konvergenssi.

Konvergenssituloksia ei tämän tutkielman puitteissa todisteta.

3.3 Marginaalijakauma

Edellä kuvattu algoritmi 1 tuottaa approksimaation koko prosessin posteriorijakaumalle $p(x_{1:k}|y_{1:k})$. Jos halutaan tietää ainoastaan posteriorijakauman $p(x_k|y_{1:k})$ estimaatti, voidaan käyttää ainoastaan viimeisestä tilasta x^i_k laskettua estimaattia

$$\hat{p}(x_k|y_{1:k}) = \sum_{i=1}^{N} w^i_{k|k} \delta(x_k - x^i_k).$$

Toinen vaihtoehto on käyttää laskennassa tärkeytyspainoa

$$w^i_{k+1|k} = \frac{\sum_{j=1}^{N} w^j_{k|k} p(x^i_{k+1}|x^j_k)}{q(x^i_{k+1}|x^i_k, y_{k+1})}$$

yllä esitetyn sijaan. Tällöin jokaisessa aikapäivitysaskeleessa lasketaan painot kaikkien mahdollisten tila-aika-avaruuspolkujen yli. Samoin kuin uudelleenotanta tämä pienentää painojen varianssia.

3.4 Aikakompleksisuus

Algoritmin perusmuodon aikakompleksisuus on $\mathcal{O}(N)$. Uudelleenotantamenetelmän tai ehdotusjakauman valinta ei suoraan vaikuta aikakompleksisuuteen. Sen sijaan marginalisointi edellä esitetyllä tärkeytyspainolla lisää algoritmin aikakompleksisuutta $\mathcal{O}(N) \rightarrow \mathcal{O}(N^2)$, koska jokaisen partikkelin kohdalla painot lasketaan jokaisen tila-aika-avaruuspolun yli. On selvää, että erityisesti isoilla otoskoon N arvoilla ei yllä esitetty marginalisointi enää ole mielekästä. Tällaisia tilanteita varten esitetään alla vielä $\mathcal{O}(N\log(N))$ versio marginaalihiukkassuotimesta. Esitys noudattaa Klaas &al. artikkelia "Toward Practical N^2 Monte Carlo: the Marginal Particle Filter" (2012). Algoritmin tehokkaaseen toteutukseen R-kielellä palataan luvussa Paikannusesimerkki.

4 Paikannusesimerkki

Esimerkissä käyteteään SMC-algoritmia Bluetooth-paikannussovelluksessa paikannustarkkuuden parantamiseen. Paikannusdata kerätään toimistoympäristössä liikkuvien BLE-lähettimien sekä kattoon sijoitettujjen vastaanottimien avulla. **kerättyjä BLE5.1-standardin mukaisia signaalin tulokulmahavaintoja eli AoA-havaintoja (angle of arrival). Paikannukseen käytetään triangulaatio-algoritmia. Lopuksi esimerkissä analysoidaan ja vertaillaan algoritmin eri versioiden suorituskykyä sekä suorituskyvyn että paikannustarkkuuden näkökulmasta.

4.1 Datan kuvaus

Riippuen päällä olevien paikannustagien lukumäärästä, tuottaa n. havainto sekunnissa. Jokainen havainto koostuu taulukossa 1 kuvatuista muuttujista.

Muuttuja	Kuvaus	Esimerkkiarvo
ts	124	124
asset_tag_mac	a	124
locator_mac	vvv	124
rssi	vvv	124
azimuth_angle	vvv	124
elevation_angle	vvv	124
quality_ant_phase_jitter	vvv	124
quality_ant_snr	vvv	124

Taulukko 1: Muuttujat

4.2 Koeasetelma

Lähettimenä toimii Lähettimenä toimi 25 Bluetooth-paikannustagista koostuva Walkbase Foculator -testilaite.

KARTTA, jossa sensorit.

TRAJECTORY. jotka MITÄ. The sensor is being built around the BLE5.1 angle of arrival (AoA) specifications, enabling it to produce sub meter positioning results.

Esimerkkidatana käytetään AoA-havaintoja seitsemän vastaanottimen sekä XXX lähettimen välillä. Dataa kerättiin. Data kerättiin yöaikaan, jolloin toimiston käyttöaste oli minimissä. Tällä minimoitiin radiosignaalin tielle osuvien ihmisten vaikutus tuloksiin.

4.3 Malli

4.4 Algoritmi

4.5 Tulokset

5 Lopuksi

Tässä tutkielmassa on esitetty pääpiirteittäin ... ja lisäksi tarkasteltu miten eri valinnat vaikuttavat algoritmin suorituskykyyn yksinkertaisen mutta XXX esimerkin avulla. Ei oteta kantaa esimerkiksi JATKOKYSYMYKSIÄ mahdollisuus laajentaa.

6 Lähteet

- Chen & Liu (1998): **Sequential Monte Carlo Methods for Dynamic Systems**. http://stat.rutgers.edu/home/rongchen/publications/98JASA_SMC.pdf
- Dahlin & Schön (2019): **Getting Started with Particle Metropolis-Hastings for Inference in Nonlinear Dynamical Models**. https://arxiv.org/pdf/1511.01707.pdf.
- Del Moral (1996): **Nonlinear Filtering: Interacting Particle Resolution**. https://people.bordeaux.inria.fr/pierre.delmoral/delmoral96nonlinear.pdf.
- Gordon, Salmond & Smith (1993): **Novel approach to nonlinear/non-Gaussian Bayesian state estimation**. http://www.irisa.fr/aspi/legland/ref/gordon93a.pdf.
- Gustafsson (2010): **Particle Filter Theory and Practice with Positioning Applications**. https://ieeexplore.ieee.org/document/5546308.
- Klaas, De Freitas, Doucet (2012): **Toward Practical N2 Monte Carlo: the Marginal Particle Filter**. https://arxiv.org/abs/1207.1396
- Petris, Petrone & Campagnoli (2009): **Dynamic Linear Models with R**. Springer.
- Särkkä (2013): **Bayesian Filtering and Smoothing.** Cambridge University Press. https://users.aalto.fi/~ssarkka/pub/cup_book_online_20131111.pdf.