



Veli-Pekka Välimaa

## **MAKSIMIVOIMALAITTEEN TOISTOLASKURIN ALGORITMIN KEHITYSTYÖ**

# **MAKSIMIVOIMALAITTEEN TOISTOLASKURIN ALGORITMIN KEHITYSTYÖ**

Veli-Pekka Välimaa  
Opinnäytetyö  
30.5.2011  
Hyvinvointiteknologian koulutusohjelma  
Oulun seudun ammattikorkeakoulu

Koulutusohjelma	Opinnäytetyö	Sivuja	+	Liitteitä
Hyvinvointiteknologian koulutusohjelma	Insinöörityö	46	+	1
Suuntautumisvaihtoehto	Aika			
Avopalvelun teknologia	30.5.2011			
Työn tilaaja	Työn tekijä			
HYTKE	Veli-Pekka Välimaa			
Työn nimi				
Maksimivoimalaitteen toistolaskurin kehitystyö				
Avainsanat				
Maksimivoima, kuntosali, toistolaskuri, algoritmi, Matlab, kiihtyvyyys, kiihtyvyyssanturi				

Insinöörityön aiheena oli kehittää ranteessa pidettävään maksimivoimalaitteeseen voimailuliikkeiden toistot tunnistava algoritmi. Laitteen aikaisempi versio toistolaskurista oli epäluotettava, koska se ei pystynyt tunnistamaan haluttua voimailuliikettä, vaan toistoksi tunnistettiin mikä tahansa tarpeeksi suuri kiihtyvyyys. Tavoitteena oli kehittää algoritmi, joka kiihtyvyyssanturin tietojen perusteella osaisi tunnistaa suoritettujen voimailuliikkeiden toistot sekä erottaa väärät liikkeet oikeista.

Algoritmin kehitystyössä käytettiin apuna langallista analogista kiihtyvyyssanturia, jonka kiihtyvyyssiedot tallennettiin AD-muuntimen ja LabVIEW-ohjelman avulla tietokoneelle. Varsinainen algoritmin kehitystyö tehtiin Matlab-ohjelmalla. Koska kyseessä oli tuotekehitystyö, ei valmiita ratkaisuja ollut olemassa. Tästä seurasi se, että monet asiat täytyi kokeilemalla todeta joko toimiviksi tai toimimattomiksi.

Lopputuloksena saatiin algoritmi, joka tietyin rajoituksin tunnistaa suoritettujen voimailuliikesarjan toistot sekä jättää huomioimatta väärät liikkeet ja liian epäpuhtaasti suoritettujen toistot.

# SISÄLTÖ

TIIVISTELMÄ .....	3
SISÄLTÖ .....	4
1 JOHDANTO .....	5
2 KÄYTETYT TERMIT JA MENETELMÄT.....	7
2.1 Kiihtyvyys .....	7
2.2 Kiihtyvyyssanturi.....	7
2.3 Summasignaali .....	9
2.4 Numeerinen derivointi Matlabilla.....	10
2.5 Kiihtyvyyden derivointi .....	10
2.6 Ristikorrelaatio .....	10
2.7 Butterworth-alipäästösuodatin ja signaalin suodatus Matlabilla.....	11
3 ALGORITMIN KEHITTÄMINEN .....	12
3.1 Käytetyt välineet ja toistolaskurin taustatietoja .....	12
3.2 Referenssisignaalin tunnistaminen ja muodostaminen .....	13
3.3 Referenssisignaalin erityispiirteet tunnistamisen kannalta .....	18
3.4 Liikesarjan toistojen tunnistaminen .....	19
4 ALGORITMIN TESTAAMINEN .....	21
4.1 Hauiskääntö.....	21
4.2 Jalkakyykky .....	25
4.3 Soutuliike .....	28
4.4 Yhdistetyn signaalin tutkiminen .....	31
4.4.1 Hauiskäännön tunnistaminen yhdistetystä signaalista.....	31
4.4.2 Jalkakyykyn tunnistaminen yhdistetystä signaalista.....	34
4.4.3 Soutuliikkeen tunnistaminen yhdistetystä signaalista.....	36
4.5 Oikein suoritetun referenssiliikkeen vaikutus toistojen tunnistamisessa.....	37
5 ALGORITMIN TOIMINNAN ARVIOINTI SEKÄ JATKOKEHITYSIDEAT .....	40
6 LOPPUSANAT.....	43
LÄHTEET.....	45
LIITE 1. Toistolaskurin algoritmin vuokaavio	

# 1 JOHDANTO

Tämän työn tilaajana toimi Oulun seudun ammattikorkeakoulun tekniikan yksikössä toimiva Hyvinvointiteknologian tutkimus- ja tuotekehityskeskus (HYTKE). HYTKE:ssä työskentelee tutkijayliopettaja Manne Hannula, laboratorioinsinööri, tekniikan yksikön opiskelijoita sekä OAMK:n asiantuntijoita. HYTKE:ssä on jatkuvasti käynnissä useita projekteja, joista osa kuuluu opintoihin liittyviin projektitöihin, osa on oppinäytetöitä ja osa tilaustöitä. Lehtori Jaakko Kaski Oulun seudun ammattikorkeakoulusta toimi tämän oppinäytetyön ohjaavana opettajana.

Kuntoilussa on ollut jo vuosia käytössä sykemittari, jolla voidaan mitata kuntoilijan sykettä ja sen perusteella voidaan harjoittelun rasittavuustaso pitää oikealla tasolla sekä laskea kalorikulutusta, suorituksen kestoaikaa ja vaikkapa sitä, onko syke rasvanpoltoalueella. Kuntosaliharjoittelu on kuitenkin luonteeltaan erilaista ja sykemittari ei sovellu moniinkaan kuntosalilla suoritettaviin harjoitteisiin.

Kuntosaliharjoittelua silmälläpitäen HYTKE on kehittänyt maksimivoimalaitteen. Laite on ranteeseen kiinnitettävä rannekellon näköinen ja kokoinen laite, jonka avulla voidaan arvioida kuntoilijan maksimivoimataso noin puolella siitä painomäärästä, jonka henkilö todellisuudessa pystyy nostamaan. Laitteen etu on siinä, että esimerkiksi yksin harjoittelevan on paljon turvallisempaa käyttää pienempiä painoja kuin todellista maksimipainoa, ja siten loukkaantumisriski pienenee. Monille kuntoilijoille oman suoritustason seuraaminen on tärkeä osa omaa harjoittelua ja voimailulajeissa maksimivoiman kehityksen seuraaminen on yleistä.

Maksimivoimalaitteessa on myös muita toimintoja kuten toistomäärälaskuri, joka pitää kirjaa tehdyn liikkeen toistojen määrästä. Laitteen toiminta perustuu kiihtyvyyssanturiin, jonka kiihtyvyystietoja analysoidaan ja niiden perusteella pystytään esimerkiksi laskemaan toistojen määrä. Laitteen ensimmäisessä versiossa oleva toistomäärälaskuri perustui kiihtyvyyssanturin kolmelta akselilta laskettavan kokonaiskiihtyvyyden arvoon. Kun kokonaiskiihtyvyys ylitti tietyn raja-arvon, tunnisti algoritmi sen toistoksi.

Algoritmin huonona puolena oli se, ettei se ottanut kiihtyvyyden suuntaa huomioon, vaan se tulkitsi toistoksi kaikkensuuntaiset kiihtyvyydet, jotka olivat tarpeeksi suuria.

Tämän työn tarkoituksena oli kehittää toistomäärälaskurissa tarvittavaa algoritmia siten, että se huomioisi liikkeiden suunnat ja estäisi virheellisten liikkeiden tunnistamisen toistoksi. Algoritmin kehittämiseen käytettiin Matlab-ohjelmaa, joka osoittautui käteväksi työkaluksi tämäntyyppisessä kehitystyössä. Toistolaskurin algoritmin vuokaavio on liitteessä 1. Koska kyseessä oli tuotekehitystyö, algoritmin lähdekoodia ei voi julkaista, vaan se on annettu tilaajan käyttöön. Valmis algoritmi pitää lopuksi kääntää C-kielelle, jotta sitä voidaan hyödyntää maksimivoimalaitteessa, mutta se ei kuulunut tämän työn aihepiiriin.

## 2 KÄYTETYT TERMIT JA MENETELMÄT

### 2.1 Kiihtyvyys

Kappaleen paikka on se piste, jossa kappale sijaitsee tietyllä ajanhetkellä tietyssä koordinaatistossa. Matka, jonka kappale on kulkenut, määritellään siten, että se on kappaleen kulkeman reitin pituus. Kappaleen hetkellinen nopeus on sen kulkema matka tietyssä aikayksikössä. Kiihtyvyys määritellään siten, että se on kappaleen nopeuden muutos tietyssä aikayksikössä. Matemaattisesti määriteltynä kiihtyvyys on nopeuden ensimmäinen- ja matkan toinen derivaatta ajan suhteen ja sen yksikkö on  $\frac{m}{s^2}$ . (1. ; 2, s. 49.)

Putoamiskiihtyvyys on maan vetovoiman aiheuttama kiihtyvyys kappaleeseen. Yleensä käytetään pyöristettyä arvoa  $9,81 \frac{m}{s^2}$ . Putoamiskiihtyvyyden merkinä käytetään pientä g-kirjainta. (2, s. 61.)

### 2.2 Kiihtyvyyssanturi

Kiihtyvyyssanturi on laite, joka tunnistaa siihen kohdistuvat kiihtyvyydet. Antureita on yksi-, kaksi- ja kolmiakselisia. Kiihtyvyyssantureiden yksi yleisimmistä kuluttajakäyttöön valmistetuista antureista on kapasitiivinen MEMS (Micro Electro Mechanical System), joka on pieni, halpa ja kestävä kiihtyvyyssanturityyppi.

Kiihtyvyyssantureita on sekä analogisia että digitaalisia. Jos analogista kiihtyvyyssanturia halutaan käyttää digitaalisessa sovelluksessa, on sille tehtävä analogisen signaalin muunnos digitaaliseksi erillisellä AD-muuntimella. Digitaalisen kiihtyvyyssanturin toimintaperiaate on muuten samanlainen kuin analogisen, mutta siinä on itsessään sisäänrakennettu AD-muunnin.

Yksiakselinen kiihtyvyyssanturi on yksinkertainen, helppo ja aikoinaan halvin anturiratkaisu ja se riittää sellaisiin sovelluksiin, joissa mitattava kohde liikkuu tasossa edestakaisin. Koska yksiakselisella kiihtyvyyssanturilla pystytään mittaamaan vain yhdessä tasossa tapahtuvaa kiihtyvyyttä, se ei ole toimiva ratkaisu sellaisiin sovelluksiin, joissa kiihtyvyyksiä pitää pystyä mittaamaan vaikkapa kolmessa eri suunnassa. Nykyään käytetään paljon kolmiakselisia kiihtyvyyssantureita, joiden avulla voidaan mitata kaikkiin suuntiin tapahtuvat kiihtyvyydet.

Antureiden hintataso on nykyään alhainen ja ne toimivat luotettavasti. Lisäksi ne ovat tarpeeksi pieniä sekä fyysisiltä mitoiltaan että virrankulutukseltaan. Edellä mainitut seikat mahdollistavat sen, että kiihtyvyyssantureiden käyttökohteiden määrä kasvaa koko ajan. Niitä käytetään nykyään puhelimissa asennon tunnistamiseen, kannettavissa tietokoneissa kiintolevyn lukupään lukitsemiseen putoamistilanteissa, autoissa turvallisuutta parantamaan, videopelien ohjaimissa sekä lukuisissa muissa sovelluksissa.

(3.)

Yksi kiihtyvyyssantureiden suurimmista ongelmista on niiden ominaisuus, että ne tunnistavat myös maapallon aiheuttaman maan vetovoiman. Paikallaan oleva kiihtyvyyssanturi näyttää aina maan putoamiskiihtyvyyden eli yhden  $g$ :n suuruista kiihtyvyyttä. Jos jokin anturin akseleista osoittaa suoraan maapallon massakeskipistettä kohden,  $g$ -voima keskittyy kokonaisuudessaan tälle yhdelle akselille ja muiden akselien tunnistama kiihtyvyys on silloin nolla. (4, s. 7.)

Jos taas yksikään akseleista ei ole kohtisuorassa maata vasten, putoamiskiihtyvyys jakautuu kaikille akseleille niiden asennon mukaan. Akselit jaetaan komponentteihin ja niiden  $g$ -voiman suuntaiset komponentit kertovat kunkin akselin osuuden putoamiskiihtyvyydestä. Jos jokin anturin akseli on  $90$  asteen kulmassa putoamiskiihtyvyyteen nähden, sen akselin ilmaisema kiihtyvyys on nolla.

Maan vetovoiman näkymisestä kiihtyvyyssanturissa on sekä hyötyä että haittaa, mutta useasti enemmän haittaa. Hyödyiksi voidaan ajatella se, että  $g$ -voiman avulla voidaan tunnistaa esimerkiksi se, missä asennossa levossa oleva kappale on. Haittapuolena taas on se, että jos kiihtyvässä liikkeessä olevan anturin asentoa ei tiedetä tarkkaan, on tällöin mahdotonta pelkän kiihtyvyyssiedon perusteella laskea anturin todellisen kiihtyvyyden suunta.



Jos kyseessä on kiinteäliikeratainen laite, jonka paikka ja asentotieto on koko ajan tarkasti tiedettävissä, voidaan siitä putoamiskiihtyvyyden aiheuttama voima vähentää pois. Putoamiskiihtyvyyden vähentäminen onnistuu esimerkiksi kiertomatriisin avulla, johon syötetään jokaista kiihtyvyytä kohden jokaisen akselin kulmatieto. Saadut arvot vähennetään jokaisesta näytteestä erikseen, jolloin jäljelle jää vain kappaleen todellinen kiihtyvyys. Jos kappaleella on epäsäännöllinen ja ennalta-arvaamaton liikerata ja asento, on asennon tunnistamiseen käytettävä jotain apuvälinettä kuten gyroskooppia, jolta tarvittava asentotieto saadaan.

### 2.3 Summasignaali

Summasignaalilla tarkoitetaan tässä työssä signaalia, joka koostuu jokaisen kiihtyvyyksien tietyn ajanhetken volttiarvojen neliöiden summasta, josta lopuksi otetaan neliöjuuri. Käytetty kaava 1 on sama kuin vektorin pituuden laskemiseen käytetty kaava. (5, s. 25.)

$$U_{tot} = \sqrt{U_x^2 + U_y^2 + U_z^2} = \text{summasignaali} \quad \text{KAAVA 1}$$

$U_x$ ,  $U_y$  ja  $U_z$  ovat kiihtyvyyssanturien akselien jännitearvot.

Tässä työssä käytetään kiihtyvyyssanturilta saatuja jännitearvoja. Jos halutaan käyttää todellisia kiihtyvyyksiä, täytyy muunnos jännitteestä kiihtyvyydeksi tehdä ennen kuin saadut arvot sijoitetaan vektorin pituuden laskemiseen käytettyyn kaavaan (kaava 1). Kiihtyvyyssanturilta saatavat jännitteet ovat aina positiivisia, mutta todelliset kiihtyvyydet voivat olla positiivisia, negatiivisia tai kiihtyvyyden arvo voi olla nolla. Tästä seuraa se, että jos muunnos tehdään vasta sen jälkeen kun kokonaisjännite on laskettu, tulokset vääristyvät ja kokonaiskiihtyvyyden arvo ei ole oikea. Toistolaskurin toimintaan sillä ei ole juurikaan merkitystä, käytetäänkö jännite- vai kiihtyvyyssarvoja.

## 2.4 Numeerinen derivointi Matlabilla

Numeerinen derivointi Matlabilla tapahtuu diff-komennolla. Siinä käsiteltävän vektorin seuraavasta arvosta vähennetään edellinen arvo, eli toisesta vähennetään ensimmäinen arvo ja kolmannesta toinen arvo jne. Saadut arvot jaetaan näytevälin pituudella. Tästä on seurauksena se, että numeerisesti derivoitu vektori on yhden näytepisteen verran lyhyempi kuin alkuperäinen vektori. Signaalin kohina vaikuttaa suuresti numeerisen derivoinnin tulokseen ja siksi se ei sovi kohinaiselle signaalille. Tässä työssä riitti pelkästään erotuksen laskeminen, eli näytevälin pituudella jakamista ei tarvittu. (6.)

## 2.5 Kiihtyvyyden derivointi

Kiihtyvyyden derivoinnilla saadaan kiihtyvyyssignaalista laskettua kiihtyvyyden muutos aikayksikköä tai näytepistettä kohden. Matemaattisesti määriteltynä kiihtyvyyden derivointi on matkan kolmas derivaatta ajan suhteen. (7.)

## 2.6 Ristikorrelaatio

Ristikorrelaatio eli liukuva pistetulo on matemaattinen tapa mitata kahden signaalin samankaltaisuutta. Toista signaalia, yleensä lyhyempää, siirretään tietty määrä ajallisesti, jolloin löydetään maksimikohta, jossa signaalit ovat lähimpänä toistensa aaltomuotoa. Tämän maksimin paikka kertoo tuon aikaviiveen suuruuden. (8.)

Normalisoidussa ristikorrelaatioissa ristikorrelaation arvot sijoittuvat välille  $\pm 1$ . Jos ristikorrelaation arvoksi saadaan 0,9 tai yli, voidaan sitä pitää hyvänä tuloksena verrattaessa kahta signaalia keskenään. (9.)

## 2.7 Butterworth-alipäästösuodatin ja signaalin suodatus Matlabilla

Alipäästösuodatin on suodatin, joka päästää läpi signaalin matalat taajuudet, mutta suodattaa pois korkeat taajuudet. Butterworth-alipäästösuodatin on yleisesti käytetty suodatintyyppi, koska siinä on tasainen vaste päästökaistalla, jonka lisäksi estokaistalla sen askelvasteet sekä jyrkkyys ovat kohtalaisen hyvät. (10, s. 10.)

Matlabilla signaalin alipäästösuodatus voidaan toteuttaa Butterworthilla seuraavasti. Aluksi mietitään tarvittavat suodatuksen arvot, kuten esto- ja päästörajataajuudet sekä sallitut vaimennukset. Buttord-käskyllä saadaan laskettua tulevan suodattimen asteluku sekä normalisoitu estotaajuuden arvo. Nämä saadut arvot sijoitetaan butter-käskyyn, jolloin saadaan kerroinvektorit A ja B. Sijoittamalla A ja B filter-käskyyn voidaan halutut signaalit alipäästösuodattaa. (11.)

Seuraavassa on esimerkki alipäästösuodatuksesta Matlabilla. Aluksi buttordilla haetaan kertoimet.

```
>> [N, Wn] = buttord(0.08, 0.2, 3, 30)
```

```
N = 4, Wn = 0.0867
```

Sen jälkeen sijoitetaan kertoimet butteriin, jolloin saadaan kaksi vektoria.

```
>> [B,A]=butter(4, 0.0867)
```

```
B = [0.0002  0.0010  0.0015  0.0010  0.0002]
```

```
A = [1.0000  -3.2893  4.1097  -2.3059  0.4895]
```

Lopuksi suodatetaan haluttu signaali.

```
>> Referenssi_suodatettu=filter(B,A,Referenssi_suodattamaton);
```

## 3 ALGORITMIN KEHITTÄMINEN

### 3.1 Käytetyt välineet ja toistolaskurin taustatietoja

Aluksi työssä tutustuttiin käytettävissä oleviin laitteisiin ja ohjelmiin. Käytettävissä oli kolmiakselinen STMicroelectronicsin valmistama LIS3L06AL- analoginen kiihtyvyysanturi sekä National Instrumentsin valmistama AD-muunnin, joka liitettiin tietokoneeseen johdolla USB-liitännän kautta. Kiihtyvyyssignaalien tallentamiseen käytettiin LabView-ohjelmistolla tehtyä ohjelmaa. Näytteenottotaajuudeksi valittiin 60 Hz, joka riittää hyvin tällaiseen sovellukseen. Algoritmin kehitystyö tehtiin Matlab-ohjelmalla, johon kiihtyvyyssignaalit ladattiin käsiteltäviksi. Huolimatta siitä, että algoritmin on tarkoitus toimia reaaliaikaisesti maksimivoimamittarissa, käytettiin kehitystyössä tallennettuja, ei-reaaliaikaisia signaaleja.

Kehitystyön tarkoituksena oli saada kehitettyä sellainen algoritmi, joka tunnistaa ainakin kolme erityyppistä vapailla painoilla tapahtuvaa kuntoiluliikettä. Aluksi liikkeiksi valittiin penkkipunnerrus, kahden käden hauiskääntö kulmatangolla sekä jalkakyykky. Tämä aiheutti kuitenkin käytännön ongelmia, koska käytetty tallennuslaitteisto ei ollut langaton ja sen siirtäminen esimerkiksi punttisalille olisi ollut erittäin hankalaa. Laitteisto sijaitsi laboratoriossa, jossa ei ollut penkkipunnerrukseen vaadittavaa laitteistoa eli punnerruspenkkiä ja painoja. Lisäksi penkkipunnerruksen ja jalkakyykyyn liikkeiden samankaltaisuudet anturin liikkeen ja asennon kannalta johtivat siihen, että testauksessa penkkipunnerrus korvattiin soutu liikkeellä. Soutu liike suoritettiin istualtaan pienillä käsipainoilla käsien liikkua vaakatasossa. Näin pystyttiin testaamaan kolme täysin erityyppistä kuntoiluliikettä. Algoritmin tavoitteena oli tunnistaa tapahtuvat toistot sekä jättää huomioimatta sellaiset liikkeet, jotka olivat liian epäpuhtaita tai eri liikkeitä kuin haluttu liike.

Edellisen algoritmin suurin ongelma oli sen liiallinen yksinkertaisuus. Algoritmi tunnistaa liikkeen kiihtyvyysanturin kaikkien akselien yhteis- eli kokonaiskiihtyvyydestä. Kun kokonaiskiihtyvyys ylitti tietyn tason, se tulkittiin toistoksi. Tämän jälkeen algoritmista oli pieni viive, jonka aikana ei hyväksytty uutta

toistoa. Kun viive oli kulunut, sen jälkeen liikkeentunnistaminen alkoi uudelleen alusta. Tämä aiheutti sellaisia ongelmia, että joskus tapahtui väärää liikkeen ja toiston tunnistamisia. Esimerkiksi sellaisessa kuntoiluliikkeessä, joka koostui ylös- ja alaspäin suuntautuvista liikkeistä, kuten jalkakyykky, saattoi algoritmi tunnistaa kyykistymisen ja nousemisen kahdeksi toistoksi, vaikka kyseessä oli vain yksi toisto. Algoritmi ei myöskään osannut tulkita yksittäisen akselin ilmaisevaa kiihtyvyyttä, joten toistoksi tulkittiin minkä tahansa suuntainen liike huolimatta valitusta liikkeestä.

Algoritmin kehittämisen kannalta oli erityisen tärkeää ymmärtää erilaisten liikkeiden ja anturin asentojen vaikutukset kiihtyvyydataan. Kiihtyvyyssanturi antaa kiihtyvyyssiedot jännitearvoina, jotka muutetaan digitaaliseen muotoon, jotta tietokoneohjelmat ymmärtävät niitä. Vaikka kiihtyvyydata muutettiin digitaaliseen muotoon, oli se yhä jännitearvoina, eikä kiihtyvyyssarvoina.

Koska algoritmin kehitystyön alkuvaiheessa ei ollut varmuutta siitä, millainen tulevan langattoman maksimivoimalaitteen suorituskyky tulee olemaan, pyrittiin siihen, ettei algoritmista tulisi liian raskas pienitehoiselle laitteelle.

### **3.2 Referenssisignaalin tunnistaminen ja muodostaminen**

Algoritmin tehtävänä on tunnistaa haluttu liike. Jotta liike voidaan tunnistaa kiihtyvyyssignaalista, pitää olla jokin kohde, referenssi, johon tutkittavaa signaalia verrataan. Aluksi piti selvittää se, voivatko referenssiliikkeet olla valmiiksi ohjelmoituja valmiiseen laitteeseen. Useiden testien jälkeen huomattiin kuitenkin, että henkilön fyysiset ominaisuudet vaikuttavat saatuun referenssisignaaliin. Käsien pituus ja paksuus sekä ryhti ja laitteenkäyttäjän omat tottumukset vaikuttavat siihen, missä asennossa maksimivoimalaite ranteessa on. Koska maan putoamiskiihtyvyys vaikuttaa kiihtyvyyssanturin toimintaan, on laitteen asennolla merkitystä anturin antamiin lukemiin. Tämä hankaloittaa valmiiden referenssiliikkeiden tallentamista laitteen muistiin. Laitteen tässä kehitysvaiheessa päädyttiin siihen, että algoritmi toimii siten, että käyttäjä itse suorittaa haluamiensa liikkeiden referenssiliikkeet. Tästä seuraa myös käytettävyyden kannalta eräs tärkeä seikka: käyttäjä voi valita, kummassa kädessä haluaa pitää laitetta sekä miten päin laite kiinnitetään valittuun ranteeseen.

Referenssiliikkeen tallentaminen laitteen muistiin on ensiarvoisen tärkeää, koska kaikki myöhemmät kyseisen liikkeen tunnistamiset tapahtuvat referenssiliikkeeseen vertaamalla. Tietokoneen avulla olisi helppoa katsoa kiihtyvyytkuvaajasta, milloin liike alkaa ja milloin loppuu. Koska laite kuitenkin sijoitetaan ranteeseen ja sen on tarkoitus toimia itsenäisesti ilman henkilökohtaista kannettavaa tai pöytätietokonetta, ei tämänkaltainen ratkaisu tule kysymykseen. Myöskään se ei onnistu, että käyttäjä painaisi laitteen nappia silloin kun liike alkaa ja toisen kerran kun liike loppuu. Se ei onnistu siksi, koska siihen tulisi liikaa häiriökiihtyvyyksiä ja esimerkiksi jalkakyykyssä tai penkkipunnerruksessa toisen käden irrottaminen heti liikkeen päätyttyä on täysin mahdotonta. Tämän vuoksi algoritmin pitäisi pystyä tunnistamaan referenssiliikkeen alkamis- ja loppumiskohdat ilman, että käyttäjän tarvitsee erikseen kertoa niitä laitteelle tai muokata saatua dataa.

Maksimivoimalaitteen käytettävyyden kannalta luontevin tapa on toteuttaa referenssiliikkeen tallennus siten, että käyttäjä painaa nappia ja asettuu liikkeen suorittamisen alkuasentoon. Noin 5–10 sekunnin kuluttua napin painalluksesta kuuluu tai näkyy jokin merkki, joka kertoo käyttäjälle sen, että tallennus käynnistyy. Merkin jälkeen käyttäjän tulisi olla noin kaksi sekuntia paikallaan, jonka jälkeen haluttu liike suoritetaan. Kun liike on suoritettu, odotetaan jälleen muutama sekunti paikoillaan, jonka jälkeen laite ilmaisee käyttäjälle tallennuksen päättymisestä.

Referenssisignaalin tunnistamisessa päädyttiin käyttämään anturilta saatujen jännitearvojen summasignaalia. Tähän ratkaisuun vaikuttivat seuraavat asiat. Jos tarkastellaan vain yhdensuuntaista kiihtyvyyttä, voi silloin jäädä havaitsematta jotain tärkeää informaatiota, joka olisi saatavilla muita kiihtyvyyssuuntia tutkimalla. Tätä ongelmaa ei esiinny silloin, kun tiedetään tarkasti suoritettavan liikkeen liikerata sekä anturin asento suhteessa käyttäjän ranteeseen. Tällaisessa tapauksessa voidaan liike tunnistaa luotettavasti vain yhtä kiihtyvyyssuuntaa tarkastelemalla, kunhan vain valitaan tarkoitukseen parhaiten sopiva kiihtyvyyssanturin akseli. Yksinkertaisimmillaan yhden akselin avulla voidaan tietty liike tunnistaa, mutta koska käytettävissä on kolme eri akselia, on järkevää käyttää niiden kaikkien tarjoamaa informaatiota.

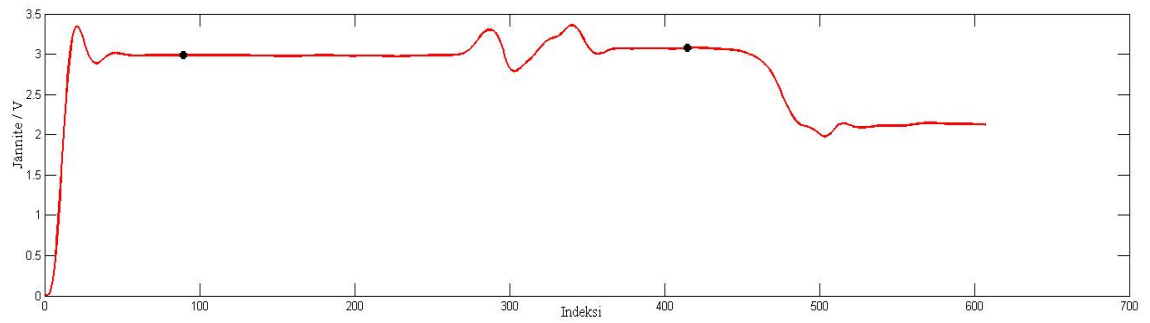
Kaikkien kolmen kiihtyvyyssanturin akseleiden käyttämisessä törmätään jälleen siihen ongelmaan, että jos niitä tarkastellaan erillisinä, pitäisi tehty liike ja anturin asento

tietää, jotta tunnistus saadaan toimimaan luotettavasti. Jos tarkastellaan vain esimerkiksi xy-tasossa tapahtuvaa liikettä, saattaa noiden kahden eri kiihtyvyyksien välillä olla pahimmassa tapauksessa isojakoin viiveitä riippuen suoritettusta liikkeestä. Tällöin ongelmaksi tulee se, kummasta signaalista tunnistetaan liike alkaneeksi ja loppuneeksi, jos suoritettusta liikkeestä ei ole tarkkaa tietoa.

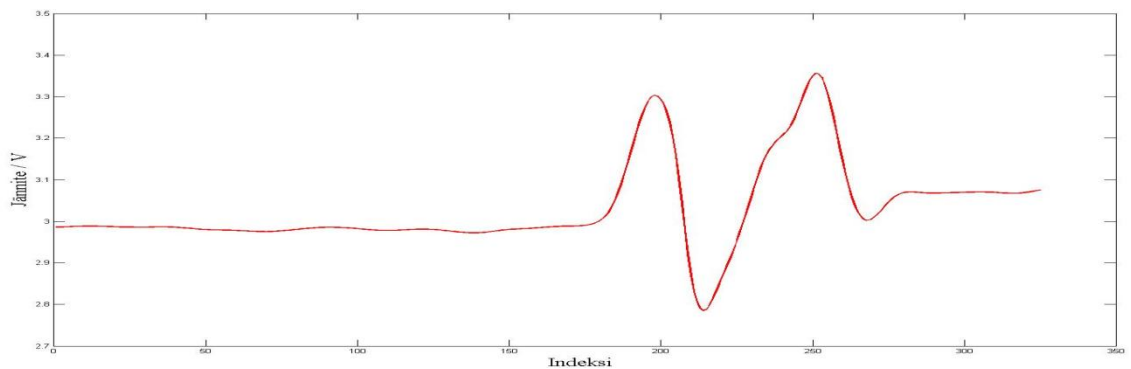
Summasignaalin käyttäminen ratkaisee edellä mainitut ongelmat, mutta täysin ongelmatonta ei summasignaalin käyttäminen ole. Summasignaalissa otetaan huomioon jokaisen akselin kiihtyvyydet ja niistä muodostetaan kokonaiskiihtyvyys kullakin ajanhetkellä. Koska kiihtyvyyksianturi reagoi myös maan putoamiskiihtyvyyteen, voi summasignaalissa kuitenkin esiintyä anturin asennonvaihteluista johtuvia kiihtyvyyden pienenemisiä tai kasvamisia, vaikka kokonaiskiihtyvyys saattaa todellisuudessa olla erisuuntaista. Summasignaalin käyttämisessä on myös se ongelma, että yhdellä akselilla saattaa olla positiivista ja negatiivista kiihtyvyyttä vuorotellen muutamia kertoja ja toisella akselilla puolestaan negatiivista ja positiivista. Pahimmillaan eri akselien kiihtyvyydet kumoavat toisiaan ja tämä tekee joissain tapauksissa summasignaalin tulkitsemisesta hankalaa. Riippuen liikkeestä sekä ranteen ja anturin asennosta voi olla, että sivuttaisliikettä ei juurikaan ole tai sitä on niin paljon, että se pitää huomioida signaaleita tutkittaessa.

Alkuperäiset signaalit olivat todella häiriöisiä johtuen kiihtyvyyksianturin tarkkuudesta. Tästä aiheutui se, että pieninkin värinä näkyi signaalissa. Häiriöistä päästiin eroon, kun kaikki käsiteltävät signaalit suodatettiin Butterworth-alipäästösuodattimella.

Käytettävissä olleessa Matlab-versiossa ei voinut suoraan valita kuvaajasta haluttua signaalin osaa, joten se täytyi ohjelmoida algoritmin alkuun, jotta erilaisten signaalien testaaminen sujuisi helpommin. Kuvassa 1 on kuvaaja referenssiliikkeen summasignaalin jännitearvoista. Vaaka-akselilla on indeksi, joka kertoo näytepisteiden lukumäärän, ja pystyakselilla on jännite voltteina. Mustien pisteiden välissä oleva kuvaajan alue on valittu tarkasteltavaksi. Tässä vaiheessa, kun käytetään Matlabia, haluttu signaalin osa täytyy valita itse. Pisteet havainnollistavat sen, milloin laite on alkanut tallentamaan kiihtyvyyksianturilta saatua tietoa ja milloin se on lopettanut. Kuvassa 2 on tallennettu referenssisignaali, jota ei ole muokattu.



*KUVA 1. Käyttäjän tekemän referenssiliikkeen kuvaaja. Tallennus tapahtuu mustien pisteiden välissä.*



*KUVA 2. Referenssiliikkeen kuvaaja ilman muokkausta*

Kuvasta 2 näkee heti sen, että siinä on liikkeeseen kuulumattomia näytepisteitä sekä alussa että lopussa. Nyt algoritmin pitäisi osata ottaa tallennetusta signaalista vain liikkeeseen kuuluva osa. Tämä tapahtuu siten, että kiihtyvyydestä derivoidaan kerran, jolloin saadaan kiihtyvyyden muutos. Kun tämä arvo pysyy määrättyjen rajojen sisällä, algoritmi tulkitsee sen lepotilaksi. Kun signaali ylittää tai alittaa asetetun ylä- tai alarajan, tulkitaan se referenssiliikkeen alkamiskohdaksi, jos näytepisteitä on tarpeeksi monta rajojen ulkopuolella. Kun tietty määrä näytepisteitä on jälleen määriteltyjen rajojen sisäpuolella, tulkitsee algoritmi silloin liikkeen päättyneeksi. Liike- tai lepotilassa olevia näytepisteitä täytyy olla tietty määrä sen takia, jotta esimerkiksi muutaman näytepisteen pituinen, rajat ylittävä jakso ei aloita tai lopeta signaalin tunnistamista. Esimerkiksi kun liike saavuttaa ääriarvonsa, pitää liikkeen pysähtyä, ennen kuin se voi muuttaa suuntaansa palatakseen alkuasentoon.



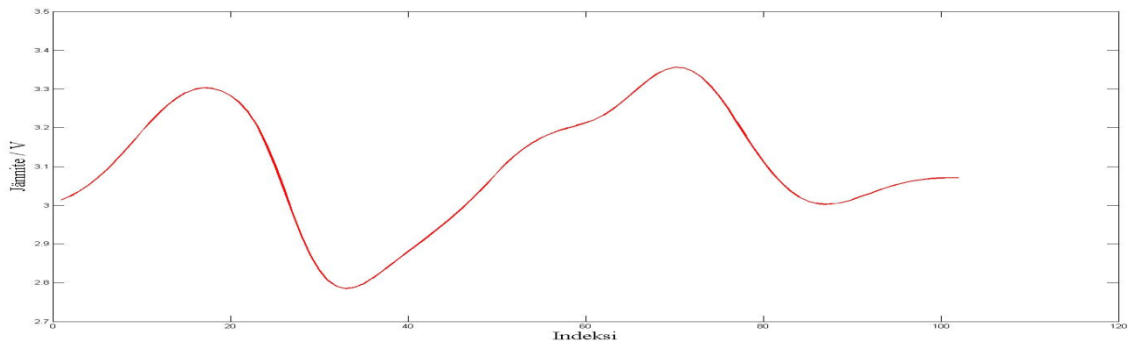
On tärkeää, että liike alkaa ja loppuu samaan kohtaan. Tämä voidaan tarkistaa siten, että verrataan referenssiliikkeen alkamis- ja loppumiskohdan anturin antamia arvoja keskenään. Jos asento on suunnilleen sama, pitäisi silloin vastaavien anturin arvojen olla hyvin lähellä toisiaan. Algoritmissa tämä tarkistus on tehty yksinkertaisesti laskemalla aloituskohdan muutaman näytepisteen jännitteen keskiarvo ja vertaamalla sitä lopetuskohdan arvoon. Jos eroa on liikaa, ehtolause antaa vastaukseksi epätosi ja käyttäjälle ilmoitetaan, että signaalin aloitus- ja lopetuskohdat eroavat liikaa toisistaan. Loppusovelluksessa luonnollisesti käyttäjää pyydetään suorittamaan referenssiliike uudestaan.

Referenssisignaalin aloituskohdan määrittämisessä pitää huomioida se seikka, että jos liipaisukynnyksenä on esimerkiksi kymmenen kappaletta lepotilan ylittäviä arvoja, pitää silloin palata ainakin kymmenen näytepistettä taaksepäin ja mieluummin vielä muutama enemmän, jotta varmasti saadaan referenssisignaalin aloituskohta oikein määriteltä. Sama pätee myös signaalin loppuosaan. Testaamalla voi selvittää, kuinka monta näytepistettä tarvitsee lisätä tai vähentää, jotta tulos on mahdollisimman hyvä.

Algoritmissa edellämainittu asia on toteutettu seuraavasti. Derivoitulle signaalille on tehty ehtolause käyttäen tai-operaattoria, joka antaa tiedon loogisina muuttujina 1 tai 0, jolloin 1 on ehdot täyttävä eli tosi ja 0 on ehdot täyttämätön eli epätosi arvo. Seuraavaksi algoritmi laskee kumulatiivisen summan saaduista loogisista arvoista siten, että summa alkaa nolasta ja jatkuu niin kauan kuin loogisia tosiarvoja on peräkkäin. Jos välissä on yksikin nolla eli epätosi arvo, alkaa kumulatiivisen summan laskeminen alusta. Kun haluttu määrä näytepisteitä on löydetty, algoritmi ilmoittaa viimeisen näytepisteen järjestysnumeron. Tästä sijainnista pitää palata taaksepäin, jotta saadaan oikea aloituskohta.

Lopetuskohdan etsiminen toimii samalla tavalla, mutta koska nyt etsitään levossa eikä kiihtyvässä liikkeessä olevia arvoja, muutetaan loogiset arvot päinvastaisiksi. Matlabissa tämä onnistuu helposti lisäämällä halutun vektorin eteen ~ -merkki. Täytyy kuitenkin huomata, että lopetuskohdan etsintä ei voi alkaa samasta kohdasta kuin aloituskohdan etsintä, koska siinä tapauksessa algoritmi tunnistaisi alkupään ehdot täyttävät kohdat ja lopetuskohta olisi väärä. Tämän välttämiseksi lopetuskohdan etsintä alkaa aikaisintaan siitä kohdasta, joka ensimmäiseksi tunnistettiin aloituskohdaksi. Lopulliseen

aloituskohtaan lisätään siis sama määrä ja muutama näytepiste enemmän kuin aloituskohdan määrittämisessä palattiin takaisinpäin. Näin voi käyttää samaa tapaa myös loppukohdan etsimisessä. Kun aloitus- ja lopetuskohta on saatu selville, voidaan muodostaa lopullinen referenssisignaali (kuva 3).



*KUVA 3. Algoritmin tunnistama referenssilieke*

### 3.3 Referenssisignaalin erityispiirteet tunnistamisen kannalta

Kun referenssisignaali on saatu muodostettua, voidaan sitä tutkia ja etsiä tiettyjä piirteitä, joita voitaisiin käyttää liikesarjan toistojen tunnistamisessa. Signaalista on helppo laskea sen sisältämien näytepisteiden määrä eli signaalin pituus, sekä paikalliset että absoluuttiset maksimi- ja minimiarvot sekä niiden esiintymisjärjestykset ja paikkatiedot.

Paikallisia minimi- tai maksimiarvoja voi olla useita. Kuitenkin vain selvimmin erottuvat arvot ovat tärkeimpiä tunnistamisen kannalta. Tämän vuoksi algoritmissa on suodatus, jolla suodatetaan sellaiset arvot pois, jotka ovat referenssisignaalin maksimiarvoihin verrattuna liian pieniä ja minimiarvoihin verrattuna liian isoja. Näin saadaan selvimmin erottuvat huiput ja laaksot mukaan, kun määritellään niin sanottuja liipaisukynnyksiä.

Liipaisukynnyksellä tarkoitetaan jotain tiettyä arvoa, jonka ylittyessä tai alittuessa tehdään jotain. Tässä algoritmissa tehdyn voimailuliikkeen referenssisignaalista määritetään liipaisukynnykset, joiden avulla liikesarjan signaalista voidaan etsiä ehdot

täyttävät kohdat. Näin vältetään turhilta laskutoimituksilta, koska jos liikesarjan signaalista ei löydy ehtoja täytettäviä kohtia, algoritmi pysyy odotustilassa.

Yleensä liipaisukynnyksiä tarvitaan useampia, jotta tunnistaminen sujuu ongelmitta. Esimerkiksi pelkän minimin tai maksimin valitseminen ainoaksi liipaisukynnykseksi voi aiheuttaa sen, että paikalliset minimi tai maksimit voivat myös täyttää liipaisukynnyksen määrittämän ehdon ja tällöin voi tulla virhetulkintoja liikkeen alkamis- tai loppumiskohdasta. Kun liipaisukynnykset on tunnistettu, ne järjestetään oikeaan järjestykseen niiden todellisen esiintymisjärjestyksen mukaan.

Referenssisignaalista saadut liipaisukynnykset eivät ole kuitenkaan vielä valmiita käytettäväksi, vaan niiden arvoja täytyy hieman muokata. Koska ei voida olettaa, että liikesarjan jokainen liike olisi tehty täysin samalla tavalla kuin referenssiliike, täytyy liipaisukynnyksiä säätää siten, että myös pienemmät kiihtyvyydet voivat laukaista liipaisukynnykset. Tämä tapahtuu siten, että maksimiarvoja pienennetään ja minimiarvoja puolestaan suurennetaan.

### **3.4 Liikesarjan toistojen tunnistaminen**

Yleensä kun suoritetaan liikesarjaa, ei toistojen välille jää välttämättä juurikaan taukoja. Tästä seuraa, että liikesarjaa analysoitaessa ja yksittäisiä toistoja tunnistettaessa ei voida käyttää samaa tapaa kuin referenssisignaalin tunnistamisessa. Tämä algoritmi käyttää apuna referenssisignaalista poimittuja liipaisutasoja. Algoritmi tutkii liikesarjan signaalia, ja kun vaadittu määrä liipaisukynnyksiä täyttää niille asetetut ehdot samanaikaisesti, siirtyy algoritmi tutkimaan kyseistä näytepisteiden sarjaa tarkemmin.

Liipaisukynnyksien avulla voidaan myös arvioida liikesarjassa esiintyvien toistojen pituutta. Referenssisignaalista voidaan laskea ensimmäisen ja viimeisen hyväksytyyn liipaisukynnyksen paikkatiedot eli etäisyydet sekä alusta että lopusta. Näiden avulla voidaan myös aloittaa liikesarjan signaalin tutkiminen hieman aikaisemmin ja lopettaa myöhemmin, kuin pelkät liipaisukynnykset edellyttäisivät. Näin voidaan varmemmin saada koko toisto vertailtavaksi, mutta vaarana on myös se, että tutkittavia näytepisteitä otetaan liikaa. Algoritmin toiminnan kannalta on ehkä parempi ottaa muutamia

näytepisteitä liian vähän kuin liian paljon, mutta käytännön testeillä pitäisi pystyä määrittämään tarpeeksi hyvä arvio tarvittavien näytepisteiden määrästä.

Seuraavaksi valitun näytepistesarjan alkuun ja loppuun lisätään referenssisignaalista laskettu määrä näytepisteitä, jotta vertailtavat signaalit olisivat suunnilleen samanmittaisia. Kun näytepistesarja on halutun pituinen, verrataan sitä referenssisignaaliin. Vertailu tapahtuu ristikorrelaation avulla.

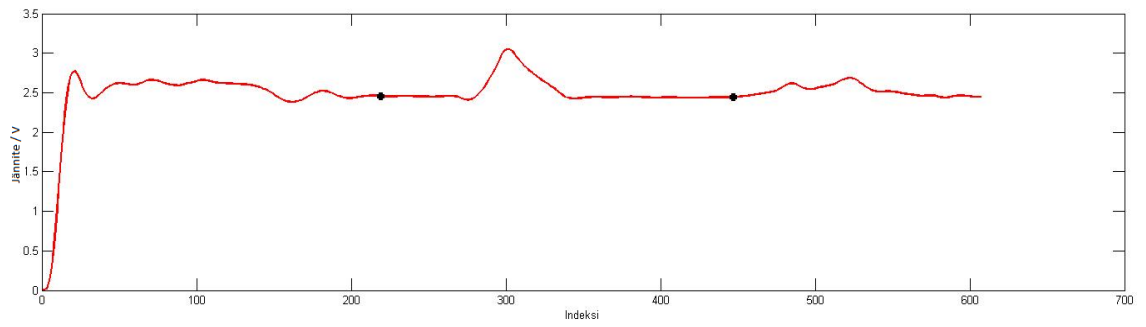
Jotta voidaan varmistua siitä, että liike on suoritettu tarpeeksi samankaltaisesti kuin referenssi, verrataan jokaista kiihtyvyyssakselia erikseen. Jos liikkeen jokaisen akselin ristikorrelaation tulokset ylittävät annetut raja-arvot, algoritmi päättelee toiston tapahtuneen ja siirtyy takaisin tutkimaan seuraavia liipaisutasoja. Tätä silmukkaa käydään niin kauan läpi, kunnes koko liikesarjan signaali on käyty läpi.

## 4 ALGORITMIN TESTAAMINEN

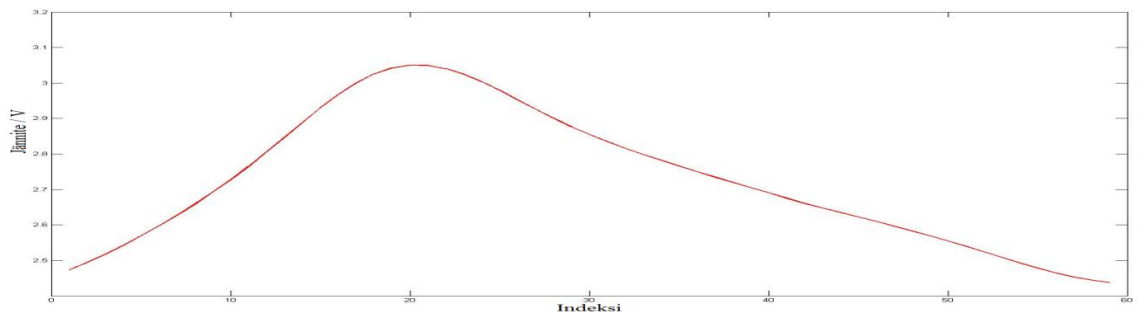
Algoritmin toiminnan testaaminen tapahtui tutkimalla erilaisia liikesarjojen kiihtyvyyssuureita. Testausta varten kiihtyvyyssanturi kiinnitettiin tukevasti muoviseen koteloon, joka oli varustettu tarranauhalla. Ranneke kiinnitettiin testikäyttäjän vasempaan ranteeseen. Laitteiston suuren koon vuoksi testimittaukset suoritettiin HYTKE:n laboratoriossa, jossa laitekokonaisuus sijaitsi. Voimailuliikkeiden suorittamisessa apuna käytettiin kulmatankoa, joka levypainoineen painoi noin 15 kg. Testikäyttäjä teki aluksi valitun voimailuliikkeen vertailukohtana toimivan referenssiliikkeen, joka tallennettiin erilliseksi tiedostoksi. Tämän jälkeen testihenkilö suoritti ennaltamäärätyn määrän liikkeen toistoja, jotka myös tallennettiin omaksi tiedostoksi. Yleensä pyrittiin saamaan mahdollisimman hyviä toistoja aikaan, mutta testausmielessä osa toistoista tehtiin huomattavasti hitaammin tai epäpuhtaammin kuin referenssiliike. Näin saatiin erityyppistä testausmateriaalia, jonka avulla voitiin arvioida algoritmin vahvuuksia ja heikkouksia.

### 4.1 Hauiskääntö

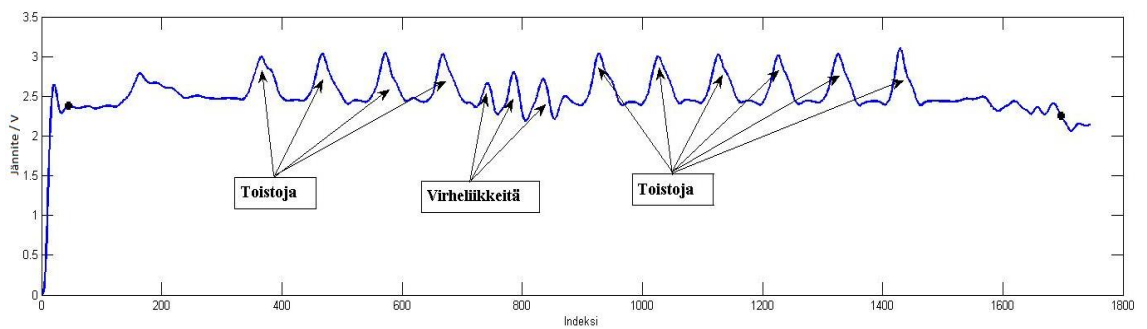
Tutkitaan aluksi hauiskääntöä. Ensimmäiseksi suoritettiin referenssiliike, joka on kuvassa 4. Kuvassa 5 on kuvattuna algoritmin valitsema referenssisignaali. Tämän jälkeen tehtiin liikesarja (kuva 6), jossa on aluksi neljä hyvää toistoa, sen jälkeen virheliikkeitä ja lopuksi vielä kuusi kappaletta hyviä toistoja eli yhteensä kymmenen hyvää toistoa. Kuvassa 7 on esitetty hauiskääntöä referenssi- ja liikesarjan signaalit samassa kuvassa.



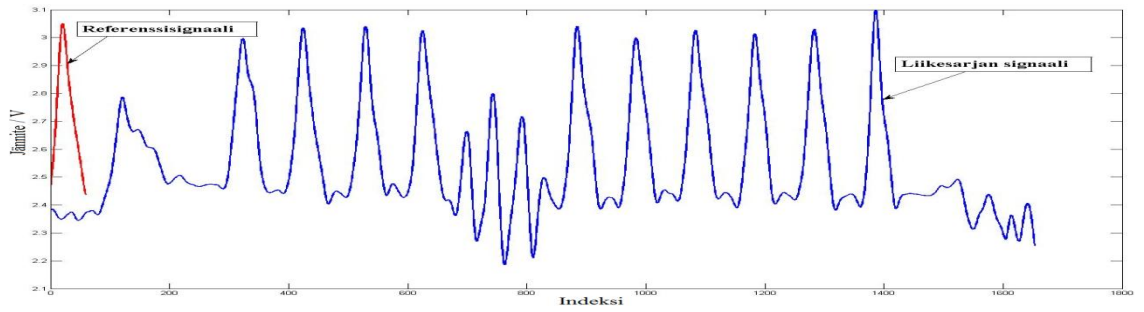
*KUVA 4. Referenssiliikkeen suorittamisen aikana tallennettu data hauiskääntöliikkeestä*



*KUVA 5. Algoritmin valitsema referenssisignaali hauiskääntöliikkeestä*

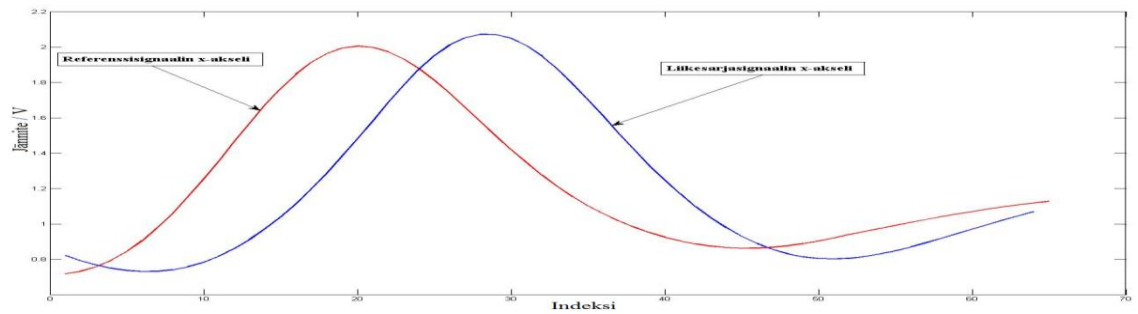


*KUVA 6. Hauiskäännön liikesarja sisältäen sekä puhtaat toistot kymmenen kappaletta että virheliikkeet*

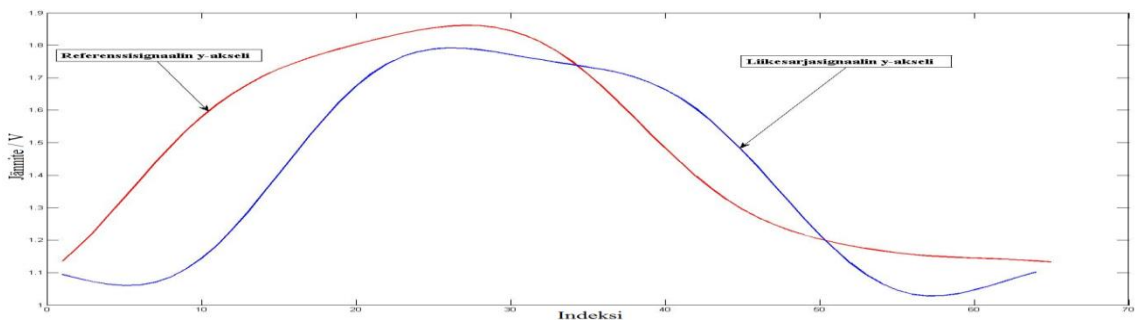


KUVA 7. Hauiskäännön referenssi- ja liikesarjan signaali samassa kuvassa

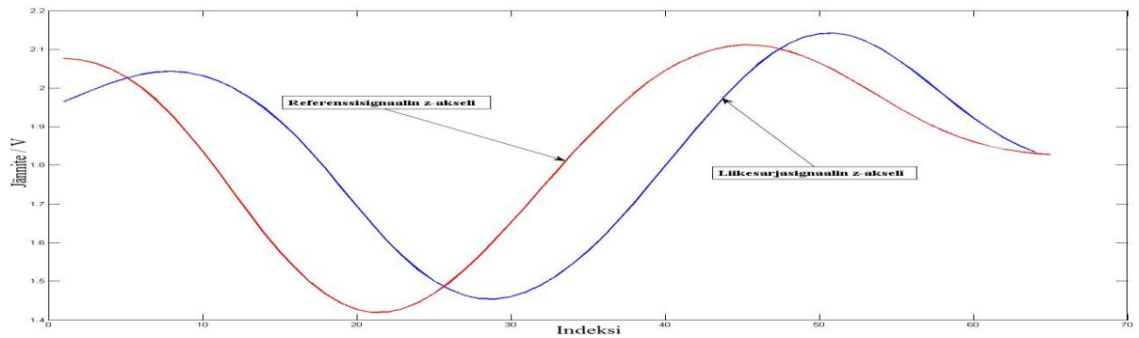
Seuraavaksi tarkastellaan viimeisen eli kymmenennen toiston x-, y- ja z-akseleiden kuvaajia referenssisignaalin vastaavien akselien kanssa. Kuvissa 8–10 on esitetty kunkin akselin referenssisignaali sekä algoritmin tunnistama kymmenes toisto liikesarjan signaalista.



KUVA 8. Referenssisignaalin x-akseli sekä liikesarjasignaalin kymmenennen toiston x-akseli hauiskääntöliikkeestä



KUVA 9. Referenssisignaalin y-akseli sekä liikesarjasignaalin kymmenennen toiston y-akseli hauiskääntöliikkeestä



KUVA 10. Referenssisignaalin z-akseli sekä liikesarjasignaalin kymmenennen toiston z-akseli hauiskääntöliikkeestä

Tässä esimerkkitapauksessa ristikorrelaation raja-arvoksi oli säädetty x- ja y- akseleille 0,9 ja z-akselille 0,85. Tämä tarkoittaa sitä, että jos toiston jonkin akselin ristikorrelaation arvo jää näiden raja-arvojen alle, kyseistä toistoa ei lasketa mukaan lopulliseen toistomäärään. Näillä asetuksilla algoritmi tunnisti kaikki kymmenen tehtyä toistoa hauisliikkeessä ja jätti huomioimatta sarjan keskellä sijaitsevat virheliikkeet. Kun z-akselin raja-arvoksi asetettiin 0,9, väheni tunnistettujen toistojen määrä vain viiteen. Tämä kertoo sen, että osa toistoista ei ollut tarpeeksi hyvin tehtyjä.

Kuvassa 11 on esitetty algoritmin suorittaman ristikorrelaatiovertailun tulokset. Tässä tapauksessa kaikkien akseleiden ristikorrelaatioiden arvot ovat reilusti yli 0,9 eli toisto on suoritettu riittävän puhtaasti. Samassa kuvassa on myös tunnistettujen toistojen kokonaismäärä eli kymmenen kappaletta.

```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Ristikorrelaatio_x =
    0.9489

Ristikorrelaatio_y =
    0.9886

Ristikorrelaatio_z =
    0.9874

TOISTOJEN_LUKUMAARA =
    10

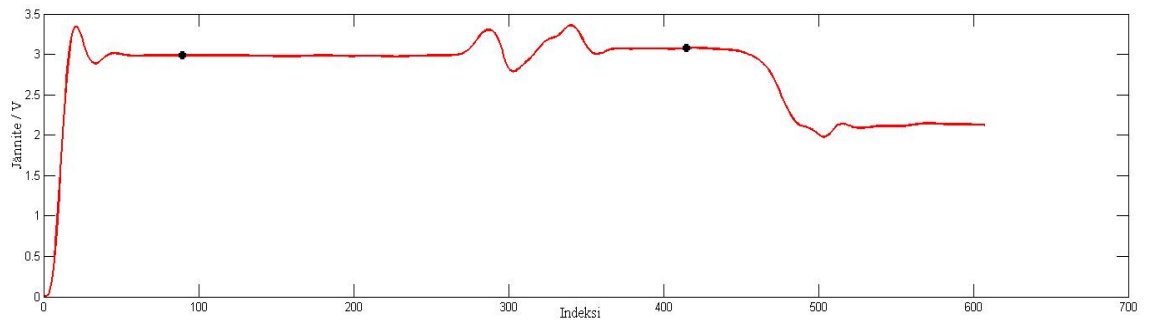
```

KUVA 11. Hauiskääntöliikkeen ristikorrelaatioiden arvot sekä tunnistettujen toistojen kokonaismäärä

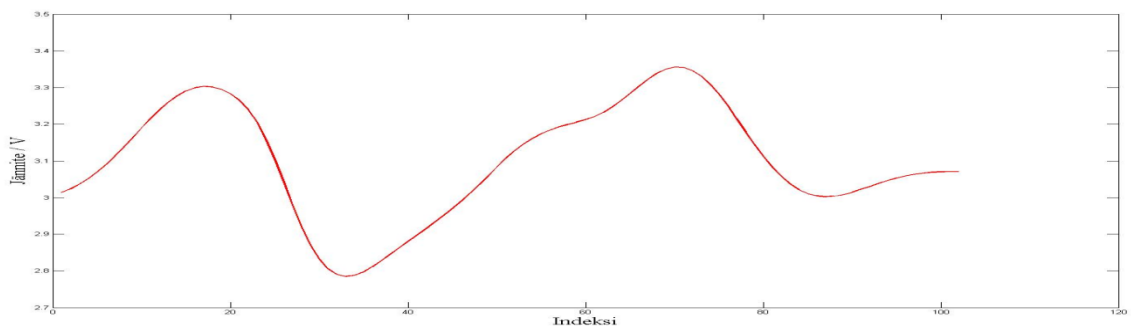


## 4.2 Jalkakyykky

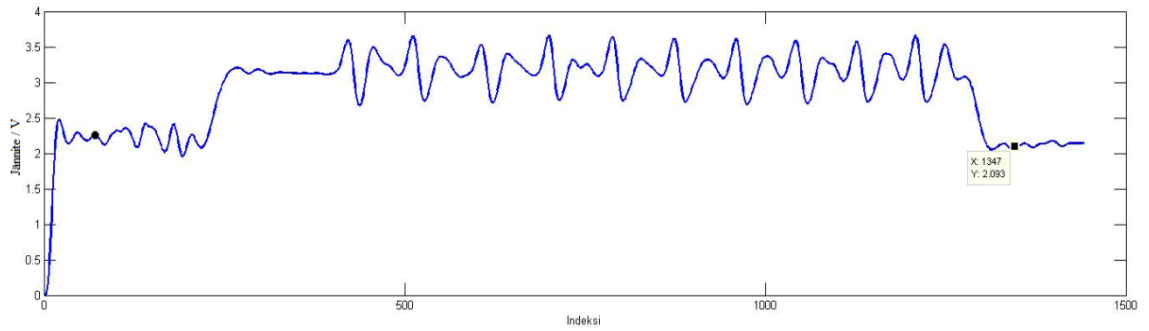
Seuraavaksi tarkastellaan jalkakyykkyä. Aluksi suoritettiin referenssiliike ja sen jälkeen liikesarja, jossa oli kymmenen toistoa. Kuvassa 12 on esitetty referenssiliikkeen aikana tallennettu data kokonaisuudessaan sekä tallennuspisteet, joiden väliltä algoritmi valitsee lopullisen referenssisignaalin. Vertailussa käytetty referenssisignaali on esitetty kuvassa 13. Kuvassa 14 on jalkakyykyn liikesarjasignaali, jossa on kymmenen toistoa. Vertailun vuoksi kuvassa 15 on sekä punaisella merkitty referenssisignaali että sinisellä merkitty liikesarjasignaali.



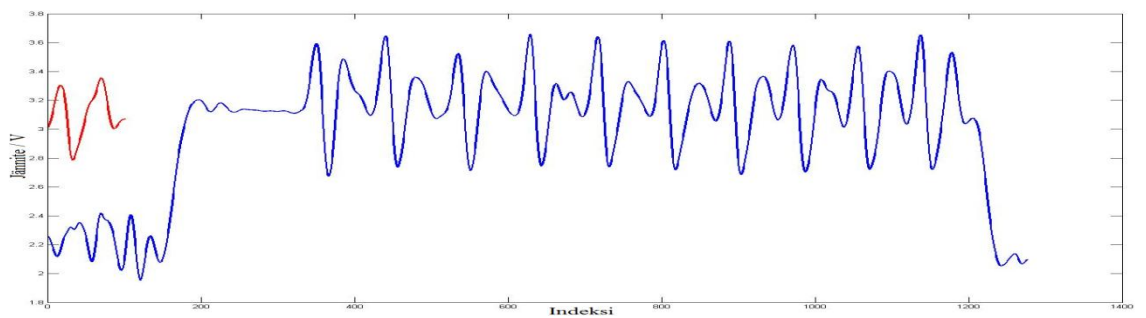
*KUVA 12. Jalkakyykyn referenssiliikkeen suorittamisen aikana tallennettu data*



*KUVA 13. Algoritmin valitsema referenssisignaali jalkakyykkyliikkeestä*

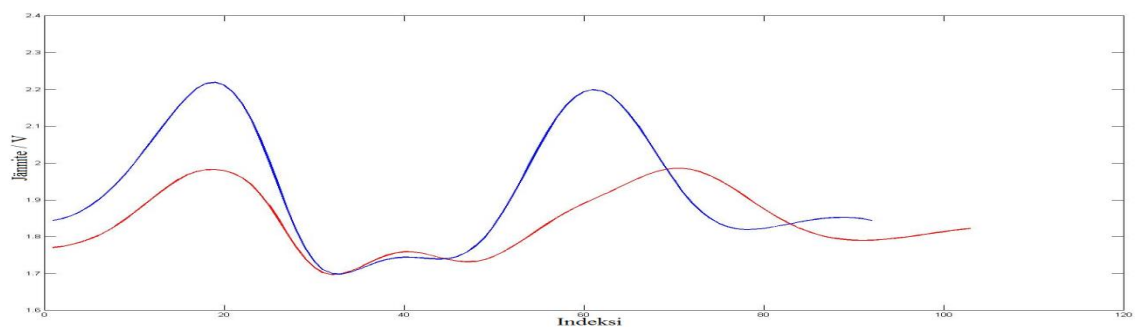


KUVA 14. Jalkakyykyn liikesarja sisältäen kymmenen toistoa

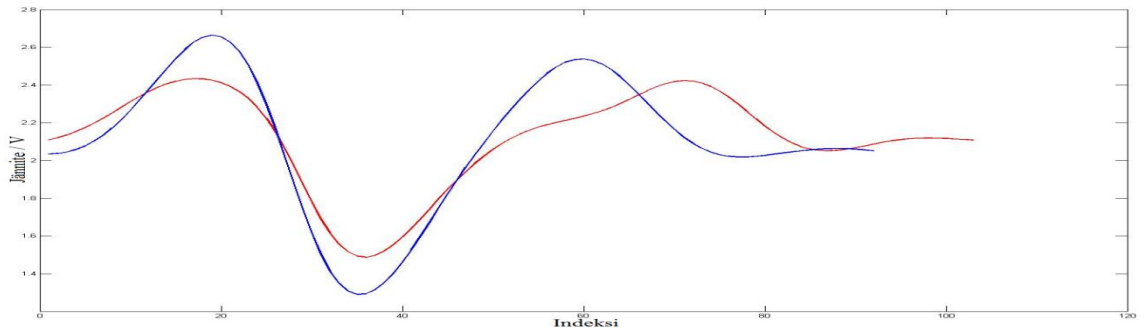


KUVA 15. Jalkakyykyn referenssi- ja liikesarjasignaali samassa kuvassa

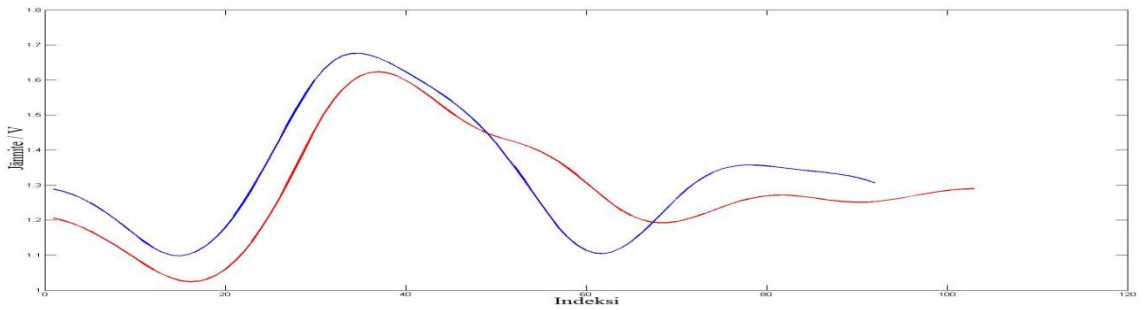
Vertaillaan yhtä yksittäistä toistoa ja referenssisignaalia. Otetaan tarkastelun kohteeksi viimeinen eli kymmenes toisto. Kuvissa 16–18 on sekä referenssisignaalin että kymmenennen toiston x-, y- ja z-akselit. Referenssisignaali on punainen ja siinä on enemmän näytepisteitä kuin sinisessä liikesarjasta poimitussa toistossa.



KUVA 16. Jalkakyykkyliikkeen referenssisignaalin x-akseli sekä liikesarjasignaalin kymmenennen toiston x-akseli



*KUVA 17. Jalkakyykkyliikkeen referenssisignaalin y-akseli sekä liikesarjasignaalin kymmenennen toiston y-akseli*



*KUVA 18. Jalkakyykkyliikkeen referenssisignaalin z-akseli sekä liikesarjasignaalin kymmenennen toiston z-akseli*

Kuvassa 19 on esitetty algoritmin suorittaman ristikorrelaatiovertailun tulokset jalkakyykylle. Samassa kuvassa on myös tunnistettujen toistojen kokonaismäärä eli kymmenen kappaletta. Ristikorrelaation raja-arvoksi oli määritetty kaikille akseleille 0,9.

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Ristikorrelaatio_x =
    0.9463

Ristikorrelaatio_y =
    0.9432

Ristikorrelaatio_z =
    0.9513

TOISTOJEN_LUKUMAARA =
    10

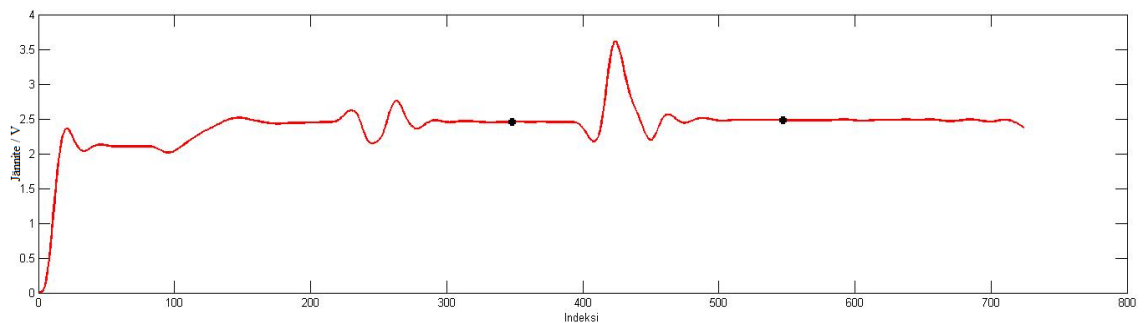
TOISTOJEN_LUKUMAARA =
    10

>>
```

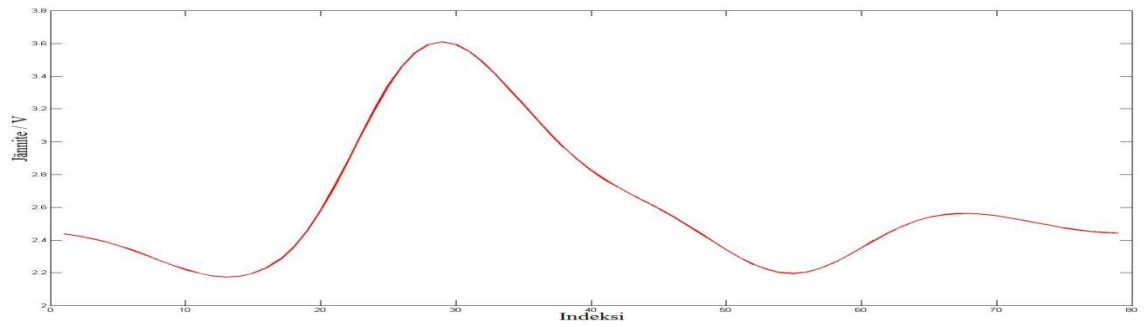
KUVA 19. Jalkakyykkyliikkeen kymmenennen toiston ristikorrelaatiot ja toistojen kokonaismäärä

### 4.3 Soutuliike

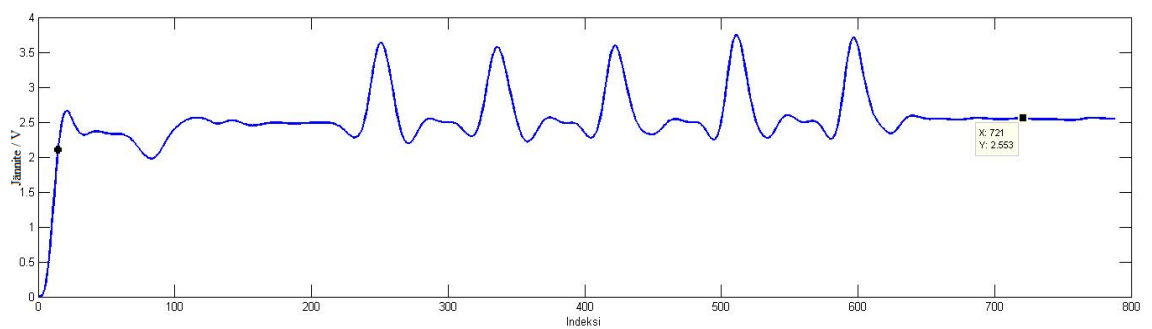
Kolmantena liikkeenä tutkitaan vaakatasossa tapahtuvaa soutuliikettä. Aluksi suoritettiin jälleen referenssiliike, joka on merkitty kuvassa 20 punaisella, ja kuvassa 21 on esitetty algoritmin valitsema referenssisignaali. Sen jälkeen suoritettiin viiden toiston liikesarja, joka näkyy kuvassa 22 sinisellä merkittynä. Kuvassa 23 on sekä referenssiliikkeen että liikesarjan signaalit samassa kuvassa.



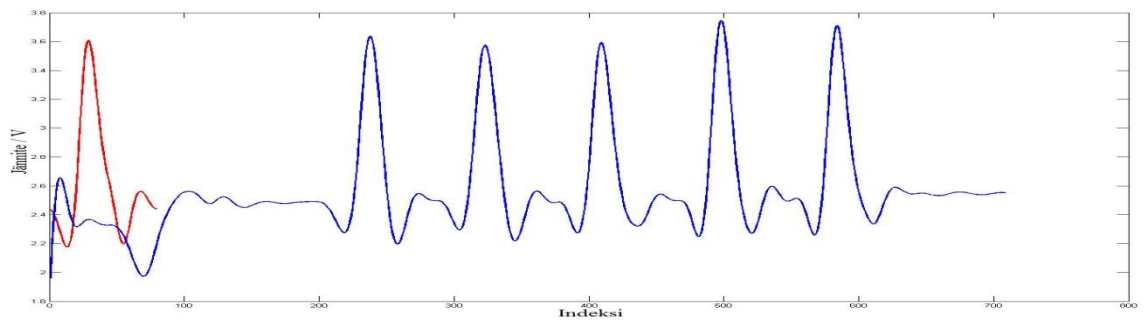
KUVA 20. Soutuliikkeen referenssiliikkeen suorittamisen aikana tallennettu data



*KUVA 21. Algoritmin valitsema referenssisignaalisuutuliikkeestä*

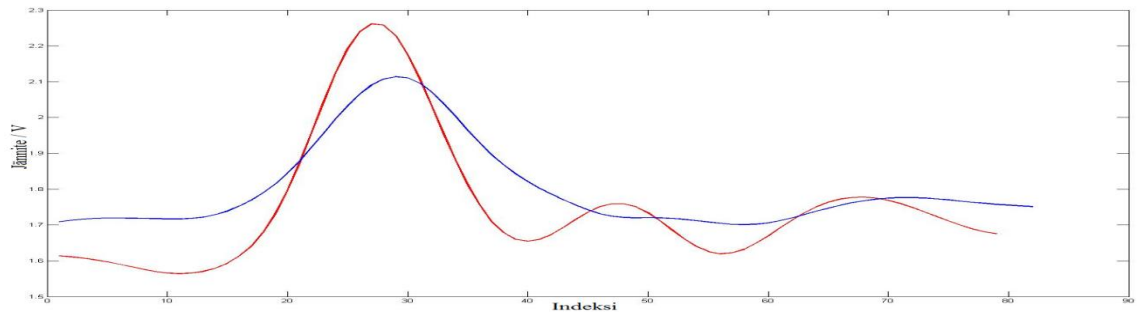


*KUVA 22. Suutuliikkeen viiden toiston liikesarja*

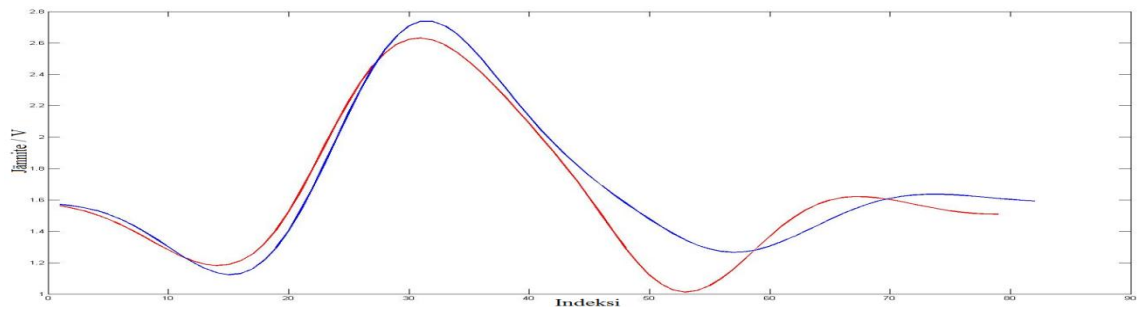


*KUVA 23. Suutuliikkeen referenssi- ja liikesarjan signaali samassa kuvassa*

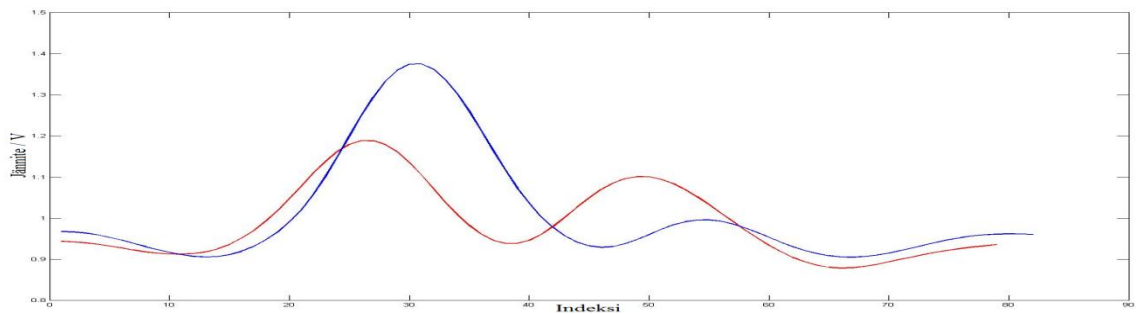
Lopuksi tarkastellaan viimeistä eli viidettä tunnistettua toistoa ja vertaillaan sen eri aksleita referenssisignaaliiin. Kuvissa 24–26 on esitetty järjestyksessä x-, y- ja z-akselit.



KUVA 24. Soutuliikkeen referenssisignaalin  $x$ -akseli sekä liikesarjasignaalin viidennen toiston  $x$ -akseli



KUVA 25. Soutuliikkeen referenssisignaalin  $y$ -akseli sekä liikesarjasignaalin viidennen toiston  $y$ -akseli



KUVA 26. Soutuliikkeen referenssisignaalin  $z$ -akseli sekä liikesarjasignaalin viidennen toiston  $z$ -akseli

Kuvassa 27 on esitetty algoritmin suorittaman ristikorrelaatiovertailun tulokset soutuliikkeelle. Samassa kuvassa on myös tunnistettujen toistojen kokonaismäärä eli viisi kappaletta. Ristikorrelaation raja-arvoksi oli määritetty kaikille akseleille 0,9.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

Ristikorrelaatio_x =
    0.9821

Ristikorrelaatio_y =
    0.9831

Ristikorrelaatio_z =
    0.9803

TOISTOJEN_LUKUMAARA =
    5

TOISTOJEN_LUKUMAARA =
    5

>>
```

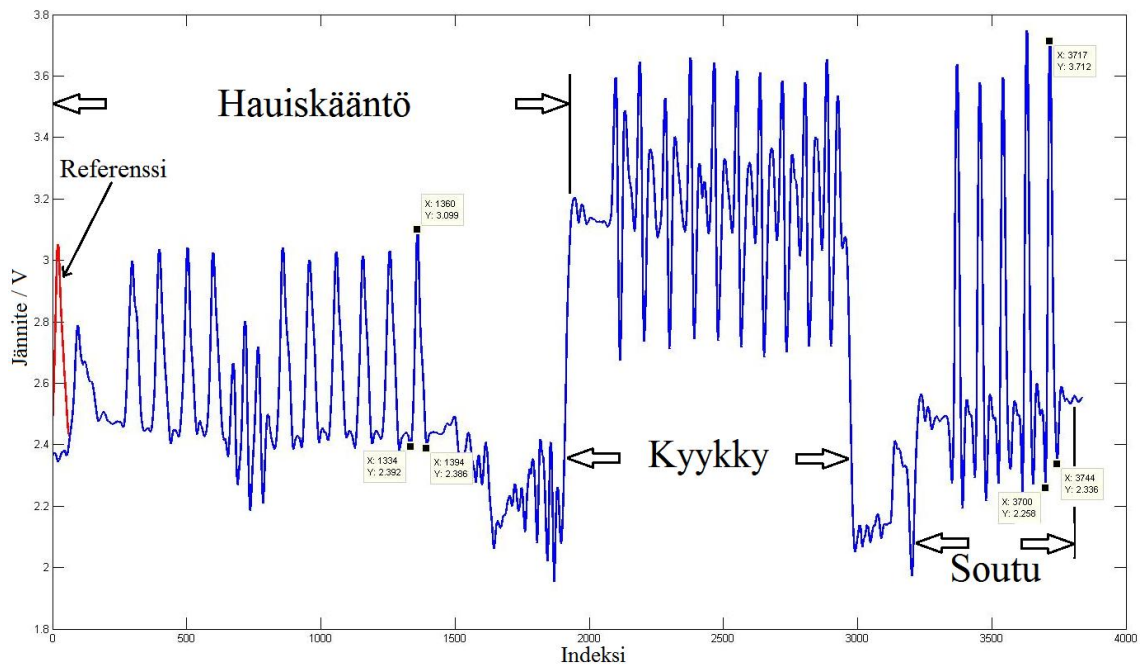
KUVA 27. Soutuliikkeen viidennen toiston ristikorrelaatiot ja toistojen kokonaismäärä

## 4.4 Yhdistetyn signaalin tutkiminen

Lopuksi tarkastellaan tilannetta, jossa kaikki kolme voimailuliikesarjaa on yhdistetty yhdeksi liikesarjaksi. Kuvassa 28 on aluksi on hauiskäännön, sen jälkeen jalkakyykyn ja lopuksi soutuliikkeen liikesarja. Kun liikesarja vaihtuu toiseksi, näkyy siinä kohdassa iso hyppäys. Eri liikkeitä on myös merkitty kuvaan. Työssä testattiin, miten algoritmi osaa erottaa oikeat toistot vääristä.

### 4.4.1 Hauiskäännön tunnistaminen yhdistetystä signaalista

Aluksi testattiin hauiskäännön tunnistaminen. Algoritmi tunnisti kaikki kymmenen toistoa oikein (kuva 29). Ristikorrelaation raja-arvoina olivat samat arvot kuin aikaisemminkin hauiskäännölle, eli arvoina oli 0,9 x- ja y-akseleille sekä 0,85 z-akselille. Lopussa olevat soutuliikkeen toistot näyttivät täyttävän liipaisuehdot, jolloin toistot etenivät ristikorrelaatiovertailuun saakka, jossa ne kuitenkin hylättiin. Kuvassa 30 on viimeisen ristikorrelaatiovertailuun hyväksytyyn liikkeen paikkatiedot sekä ristikorrelaatioiden arvot. Kun ristikorrelaation raja-arvoksi säädettiin 0,8 kaikille akseleille, oli tunnistettuja toistoja tällöin 15 kappaletta, eli viisi soutuliikettä tunnistettiin hauiskäännöiksi. Kuvissa 31–32 on esitetty viimeinen soutuliikkeen toisto, joka tunnistettiin väärin.



KUVA 28. Yhdistetty signaali, johon on merkitty viimeisen tunnistetun hauiskäännön paikkatiedot sekä viimeinen toisto, jolle tehtiin ristikorrelaatiovertailu. Kuvaan on myös merkitty siinä esiintyvät liikkeet sekä niiden järjestys.

```

Ristikorrelaatioon hyväksytyt paikat
hyvaksytty_xcorr_paikat =
    1334
    1360
    1394

Ristikorrelaatio_x =
    0.9489

Ristikorrelaatio_y =
    0.9886

Ristikorrelaatio_z =
    0.9874

TOISTOJEN_LUKUMAARA =
    10

```

KUVA 29. Viimeisen tunnistetun hauiskäännön paikkatiedot sekä ristikorrelaatioiden arvot



```

Ristikorrelaatioon hyväksytyt paikat
hyvaksytty_xcorr_paikat =
    3700
    3717
    3744

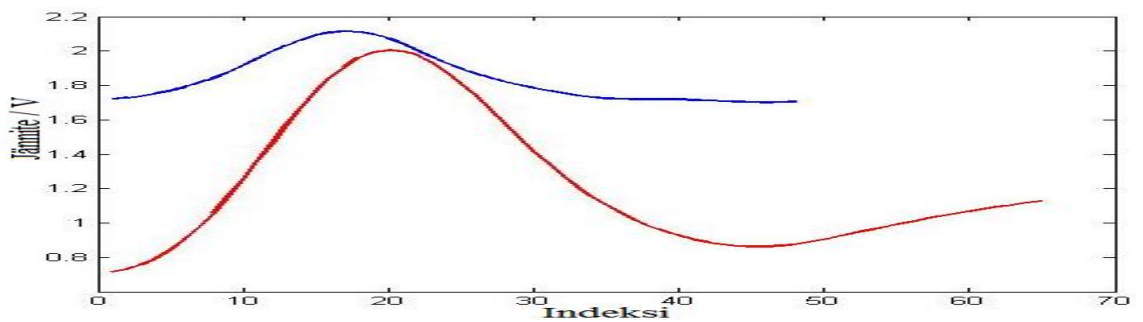
Ristikorrelaatio_x =
    0.8961

Ristikorrelaatio_y =
    0.9050

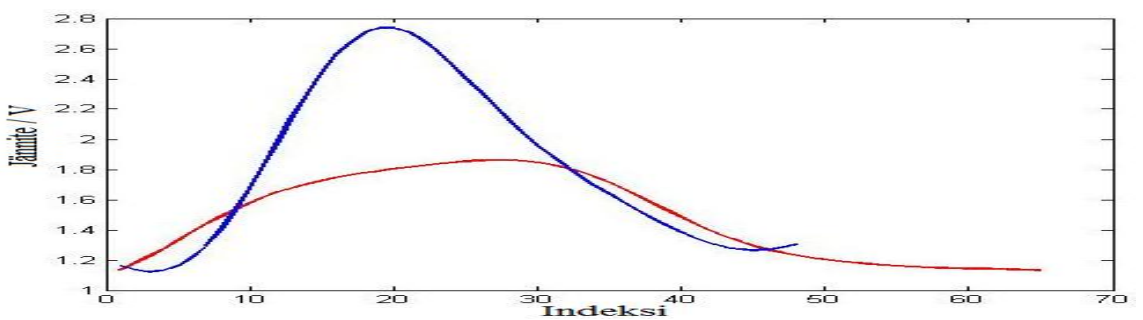
Ristikorrelaatio_z =
    0.8456

```

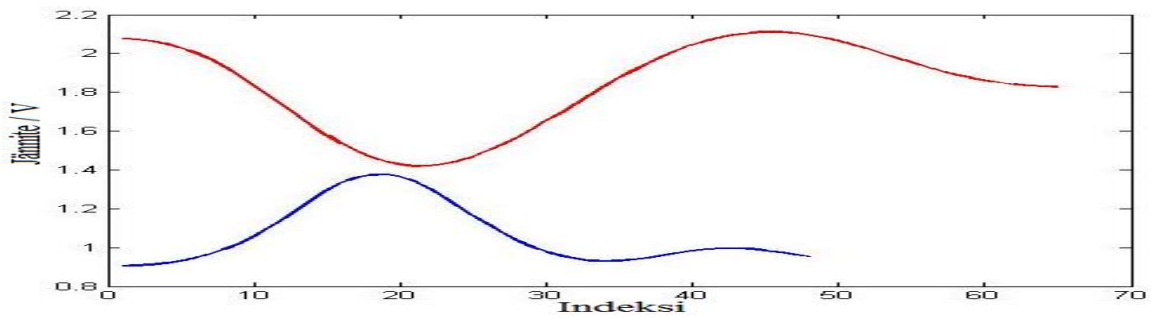
KUVA 30. Viimeisen ristikorrelaatiovertailuun hyväksytyin liikkeen paikkatiedot ja ristikorrelaatioiden arvot



KUVA 31 . Virheellisesti hauisliikkeeksi tunnistettu soutuliike, x-akseli



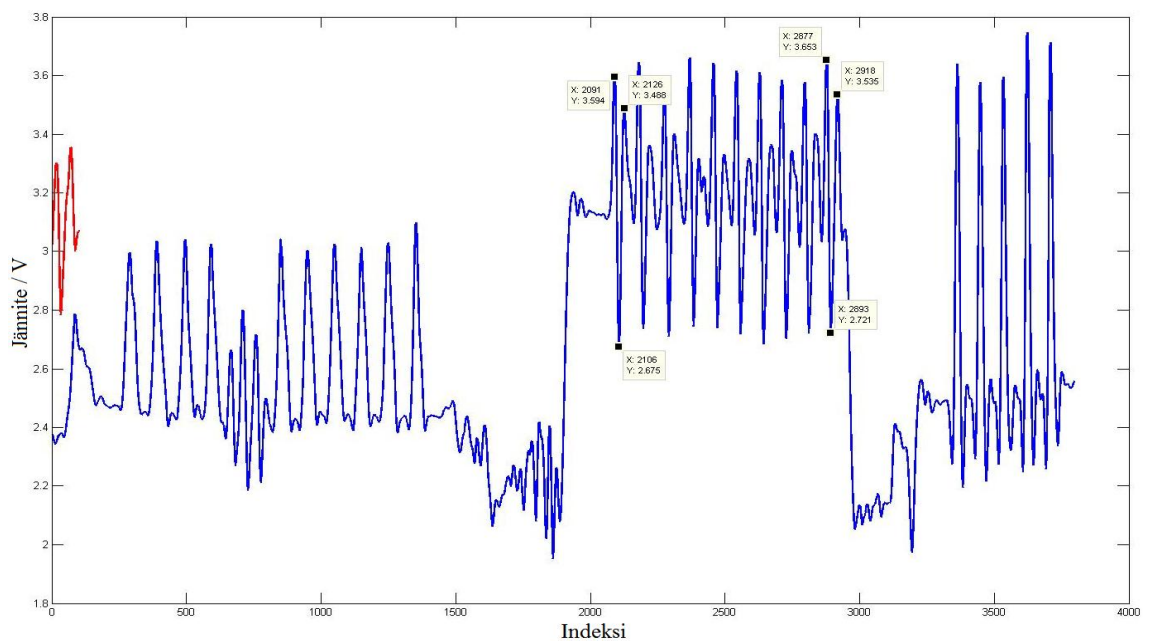
KUVA 32. Virheellisesti hauisliikkeeksi tunnistettu soutuliike, y-akseli



KUVA 33. Virheellisesti hauisliikkeeksi tunnistettu soutuliike, z-akseli

#### 4.4.2 Jalkakyykyn tunnistaminen yhdistetystä signaalista

Seuraavaksi tutkittiin jalkakyykyn tunnistamista samasta yhdistetystä signaalista. Ristikorrelaation raja-arvoksi oli asetettu kaikille akseleille 0,9. Kuvassa 33 on esitetty referenssisignaali sekä yhdistetty signaali. Siihen on myös merkitty ensimmäisen ja viimeisen tunnistetun jalkakyykyn paikkatiedot. Kuvista 34 ja 35 käyvät ilmi ensimmäisen ja viimeisen tunnistetun toiston paikkatiedot ja ristikorrelaatioiden arvot sekä toistojen kokonaismäärä.



KUVA 33. Yhdistetty signaali, johon on merkitty ensimmäisen ja viimeisen tunnistetun jalkakyykyn paikkatiedot

```

Ristikorrelaatioon hyväksytyt paikat
hyvaksytty_xcorr_paikat =
    2091
    2106
    2126

Ristikorrelaatio_x =
    0.9166

Ristikorrelaatio_y =
    0.9135

Ristikorrelaatio_z =
    0.9289

TOISTOJEN_LUKUMAARA =
    1

```

*KUVA 34. Ensimmäisen tunnistetun jalkakyykyn paikkatiedot ja ristikorrelaatioiden arvot*

```

Ristikorrelaatioon hyväksytyt paikat
hyvaksytty_xcorr_paikat =
    2877
    2893
    2918

Ristikorrelaatio_x =
    0.9457

Ristikorrelaatio_y =
    0.9425

Ristikorrelaatio_z =
    0.9520

TOISTOJEN_LUKUMAARA =
    10

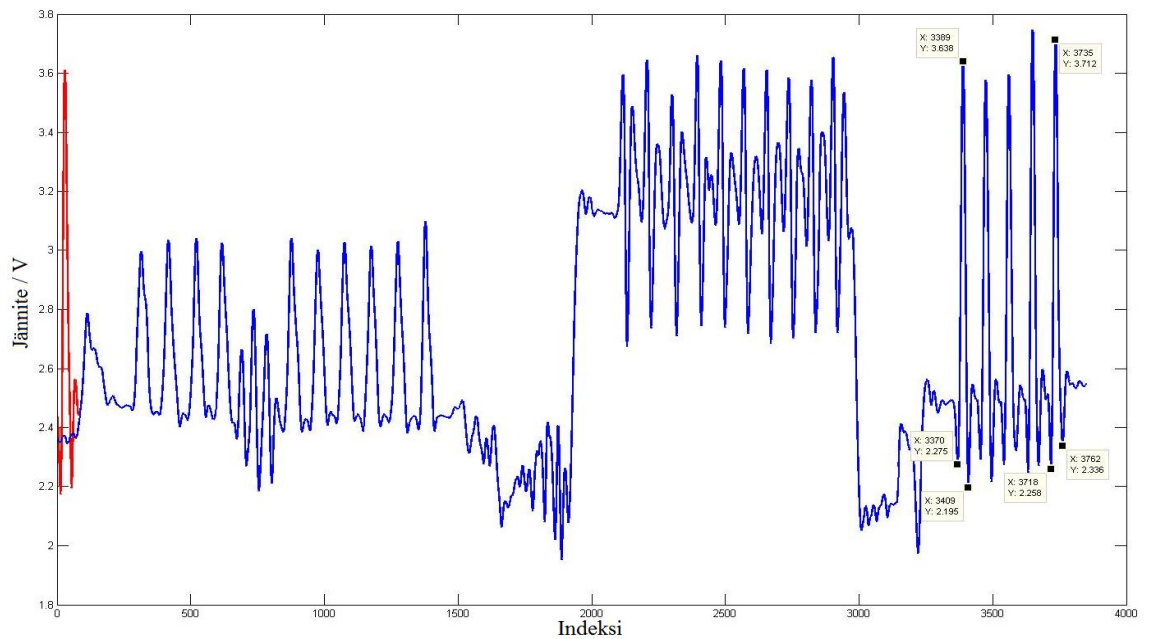
```

*KUVA 35. Viimeisen tunnistetun jalkakyykyn paikkatiedot ja ristikorrelaatioiden arvot*

Algoritmi tunnisti kymmenen toistoa, ja koska tässä tapauksessa liipaisutasot olivat huomattavasti poikkeavia verrattuna hauiskääntöön ja soutuliikkeeseen, ei tällä kertaa tehty turhia ristikorrelaatiovertailuja. Koska turhia vertailuja ei tehty, ei ristikorrelaation raja-arvon laskeminen 0,1:een vaikuttanut tulokseen mitenkään. Sen sijaan raja-arvon nostaminen 0,95:een vähensi tunnistettujen toistojen määrää vain kahteen kappaleeseen.

#### 4.4.3 Soutuliikkeen tunnistaminen yhdistetystä signaalista

Lopuksi tutkittiin soutuliikkeen tunnistamista yhdistetystä signaalista. Ristikorrelaation raja-arvoksi asetettiin kaikille akseleille 0,9. Kuvassa 36 on soutuliikkeen referenssisignaali sekä yhdistetty signaali. Kuvissa 38 ja 39 on ensimmäisen ja viimeisen tunnistetun toiston paikkatiedot sekä ristikorrelaatioiden arvot.



KUVA 36. Yhdistetty signaali, johon on merkitty ensimmäisen ja viimeisen tunnistetun soutuliikkeen paikkatiedot

```
Ristikorrelaatioon hyväksytyt paikat
hyvaksytty_xcorr_paikat =
    3370
    3389
    3409

Ristikorrelaatio_x =
    0.9848

Ristikorrelaatio_y =
    0.9875

Ristikorrelaatio_z =
    0.9868

TOISTOJEN_LUKUMAARA =
    1
```

KUVA 37. Ensimmäisen tunnistetun soutuliikkeen paikkatiedot ja ristikorrelaatioiden arvot

```

Ristikorrelaatioon hyväksytyt paikat
hyvaksytty_xcorr_paikat =
    3718
    3735
    3762

Ristikorrelaatio_x =
    0.9821

Ristikorrelaatio_y =
    0.9831

Ristikorrelaatio_z =
    0.9803

TOISTOJEN_LUKUMAARA =
    5

```

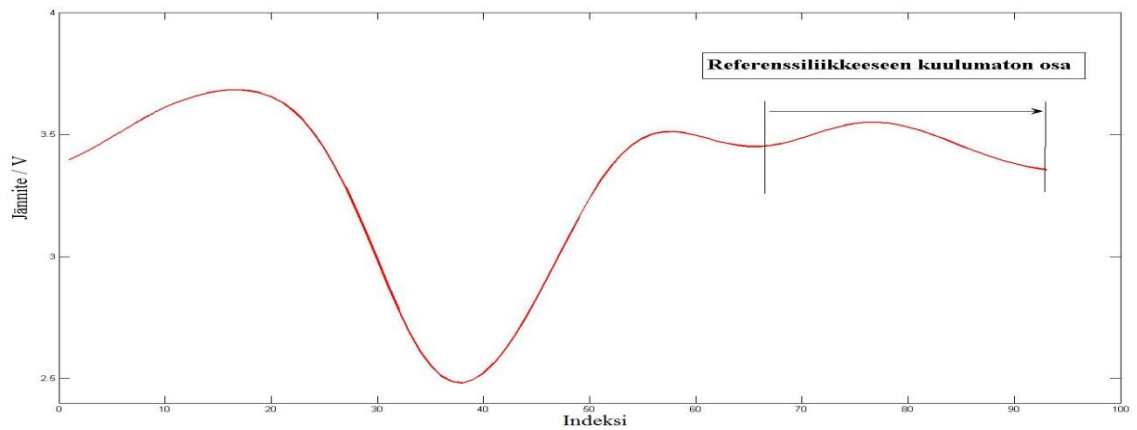
*KUVA 38. Viimeisen tunnistetun soutu liikkeen paikkatiedot ja ristikorrelaatioiden arvot*

Algoritmi tunnisti viisi kappaletta soutu liikkeitä yhdistetystä signaalista eli juuri oikean määrän. Myös tässä tapauksessa liipaisukynnykset estivät aiheettomat ristikorrelaatiovertailut, eli raja-arvon laskeminen ei vaikuttanut tulokseen. Myöskään raja-arvon nostaminen niinkin korkeaksi kuin 0,97 ei vaikuttanut tunnistettujen toistojen määrään. Sen sijaan raja-arvon nostaminen lähes maksimiin eli 0,985:een vähensi toistojen määrän kolmeen kappaleeseen.

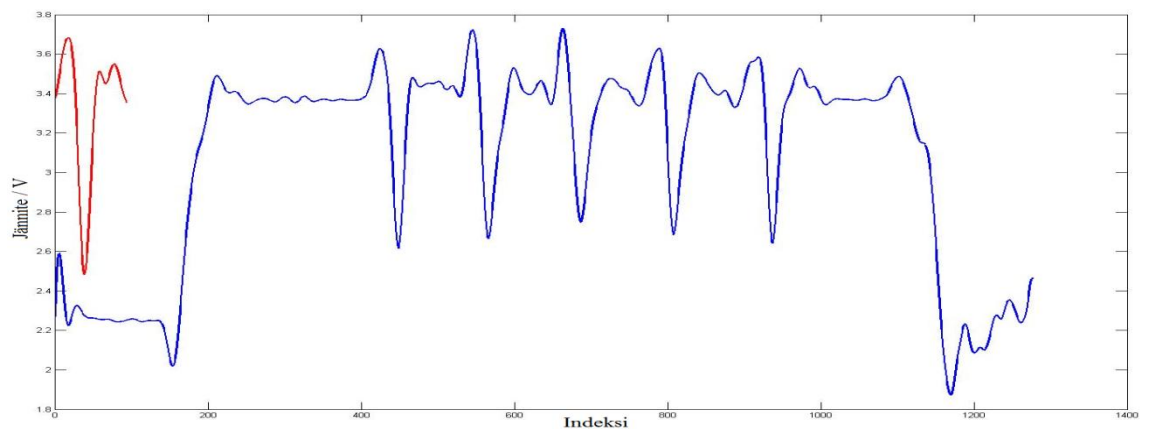
#### **4.5 Oikein suoritetun referenssiliikkeen vaikutus toistojen tunnistamisessa**

Kuten jo monta kertaa on käynyt ilmi, algoritmin luotettavuuteen vaikuttaa erityisen paljon referenssiliikkeen oikea suoritustapa. Kuvassa 39 on referenssiliike istuen suoritettuna liikkeestä, jossa kädet ovat alussa taitettuna pään yli taaksepäin ja kulmatanko on niskan takana. Tästä aloitusasennosta kulmatanko on nostettu pään yläpuolelle ja palautettu jälleen aloitusasentoon. Koska liike rasittaa käsivarren ojentajia, kutsuttakoon kyseistä liikettä tässä ojentajaliikkeeksi.

Ongelma tässä referenssiliikkeessä on se, että sen loppuun on tullut pieni heilahdus, joka on otettu mukaan lopulliseen referenssiin. Tämän seurauksena kyseisellä referenssillä algoritmi tunnisti vain yhden toiston viiden toiston ojentajaliikkeen liikesarjasta. Kuvassa 40 on sekä väärin tehty referenssi että ojentajaliikkeen liikesarja.



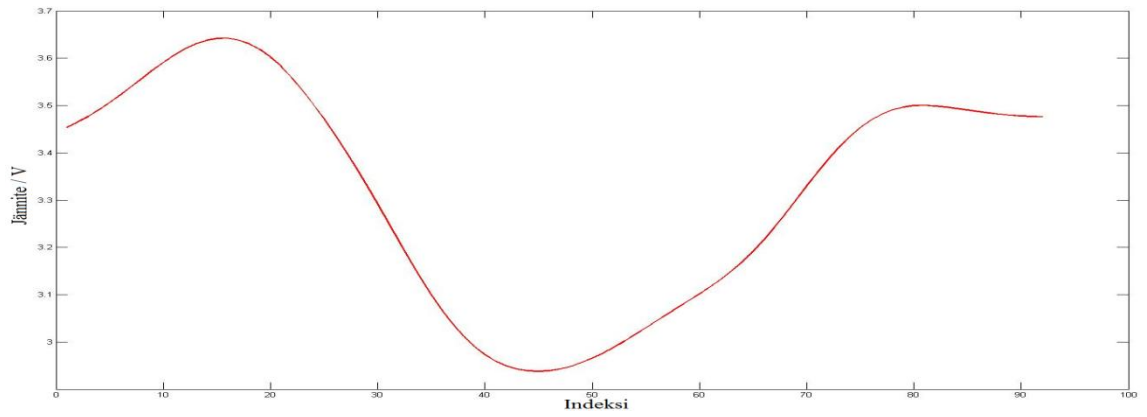
KUVA 39. Ojentajaliikkeen epäonnistunut referenssiliike



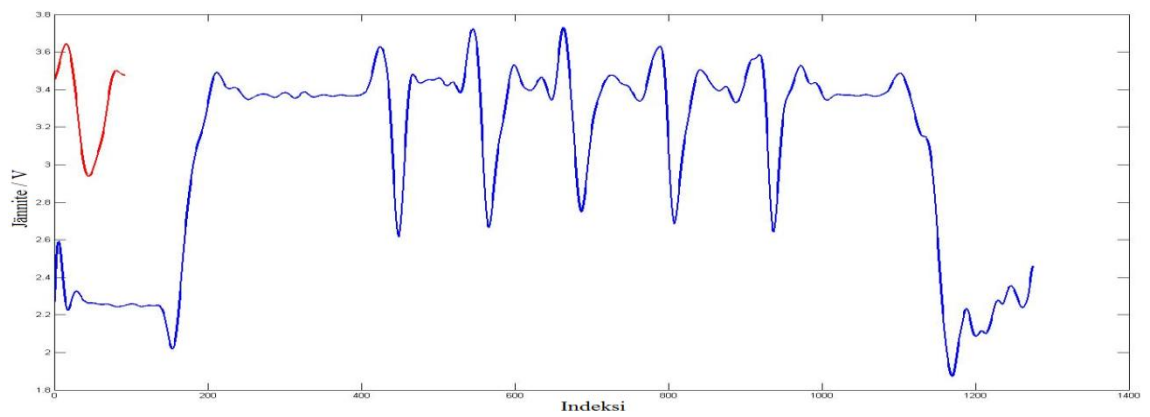
KUVA 40. Epäonnistunut referenssiliike sekä ojentajaliikkeen viiden toiston liikesarja

Referenssiliike on suoritettu hieman huolimattomasti ja tästä on seurauksena se, että referenssiliikkeen loppuun on tullut muutama heilahdus. Vaikka liikesarjassakin on epäpuhtauksia eli liikkeeseen kuulumattomia heilahduksia, ei algoritmi kuitenkaan tunnistanut kuin yhden toiston viiden sarjasta.

Kun sama liike suoritettiin uudelleen ja tehtiin uusi referenssiliike (kuva 41), jäivät referenssiliikkeen lopusta turhat värähtelyt pois ja sen myötä algoritmin toiminta parantui. Kuvassa 42 on uusi referenssiliike sekä viiden toiston ojentajaliikkeen liikesarja.



*KUVA 41. Puhtaasti suoritettu ojentajaliikkeen referenssiliike*



*KUVA 42. Onnistunut referenssiliike sekä ojentajaliikkeen viiden toiston liikesarja*

Kun referenssiliike oli puhtaammin suoritettu, algoritmi tunnisti sen avulla kaikki viisi toistoa ojentajaliikkeen liikesarjasta. Tästä esimerkistä käy hyvin ilmi, että tämän algoritmin luotettavuus perustuu paljon referenssiliikkeen oikeaan suoritustapaan. Vaikka tässä tapauksessa itse liikesarja ei ole täydellisesti suoritettu ja siinä on myös ylimääräisiä heilahteluita toistojen välillä, ne eivät ole toistolaskurin luotettavuuden kannalta niin merkittäviä.

## 5 ALGORITMIN TOIMINNAN ARVIOINTI SEKÄ JATKOKEHITYSIDEAT

Kehitystyön kohteena olleen algoritmin toiminnan luotettavuuteen vaikuttaa monta asiaa. Yksi tärkeimmistä asioista on se, että referenssisignaali saadaan tallennettua juuri oikeanlaisena. Tähän puolestaan vaikuttavat seuraavat seikat. Referenssisignaalin alku- ja loppupisteiden tunnistaminen pitää tapahtua oikein. Ne tunnistuvat oikein siinä tapauksessa, että laite on paikoillaan, samassa asennossa ja kohdassa sekä ennen liikkeen alkamista että liikkeen loputtua. Toinen tärkeä asia on se, että liike suoritetaan suunnilleen samalla nopeudella kuin tulevat toistot tullaan tekemään.

Tottuneelle kuntosaliharjoittelijalle on tuskin kovinkaan haastavaa valita sellaiset painot käytettäväksi, joita jaksaa pitää paikoillaan muutaman sekunnin käsien tärisemättä, ja silti suorittaa referenssiliike suunnilleen toistoissa käytetyllä nopeudella. Periaatteessa hyvällä ohjeistuksella ongelmia ei pitäisi syntyä myöskään aloittelijoiden keskuudessa. Ongelma on kuitenkin monesti siinä, että jos käyttäjä ei tunne tarkasti laitteen toimintaa, voi virheellisiä suoritustapoja esiintyä ja nämä puolestaan johtavat algoritmin toimimattomuuteen. Algoritmi toimii kohtuullisen luotettavasti silloin, kun liikkeet tehdään mahdollisimman tarkasti ja liikeradat sekä toistojen suoritussopeudet ovat mahdollisimman samankaltaiset verrattuna referenssisignaaliin.

Summasignaalin käytön sijaan tai rinnalla voisi kokeilla käyttää jokaisen kiihtyvyyksensä erillistä signaalia. Referenssisignaalia tutkimalla voitaisiin ottaa sellaisen kiihtyvyyksensä signaali referenssiksi, jossa on tunnistamisen kannalta kaikkein selvimmät piirteet. Näin voitaisiin ehkä välttyä sellaisilta tilanteilta, joissa kahden eri liikkeen summasignaalit ovat suunnilleen samankaltaisia, mutta siltikin ne koostuvat erilaisista komponenteista. Joissain tapauksissa tällainen liike saatetaan tulkita ristikorrelaatiossakin tarpeeksi hyväksi, jolloin se voidaan virheellisesti lukea toistoksi.

Kuntosaliharjoittelussa pyritään yleensä tekemään liikkeet mahdollisimman samankaltaisesti ja puhtaasti, mutta väsymys aiheuttaa kuitenkin tiettyjä asioita



toistoihin. Lihasten väsyessä lihasten täriseminen saattaa lisääntyä, koordinaatio heikentyä sekä liikkeet hidastua. Vaikka ulkopuolisen tarkkailijan mielestä liikkeet ovat täysin tunnistettavia ja ihan hyväksytyissä suoritusrajoissa, voi kiihtyvyyssanturin tuottama data olla kuitenkin aika lailla poikkevaa verrattaessa ensimmäistä ja viimeistä toistosarjaa keskenään.

Pelkästään liikkeen hidastuminen saattaa aiheuttaa putoamiskiihtyvyyden vuoksi summasignaaliin ylimääräisiä tai puuttuvia paikallisia maksimi- tai minimiarvoja. Siitä aiheutuu se, että jopa pienet muutokset signaalin muodossa saattavat haitata liikkeen tunnistamista. Hitaan liikkeen amplitudi myös yleensä pienenee sekä samalla näytepisteiden määrä kasvaa liikkeen kestäessä pidempään. Tästä saattaa seurata, että ennalta määritellyt liipaisutasot voivat aiheuttaa sen, että liike jää kokonaan tunnistamatta, koska amplitudi ei täytä liipaisukynnyksiä.

Toinen tärkeä asia on se, että hitaassa toistolikkeessä on huomattavasti enemmän näytepisteitä kuin referenssiliikkeessä. Tämä aiheuttaa sen, että yksinkertaistettu tapa lisätä referenssisignaalista laskettu kappalemäärä näytepisteitä toiston alkuun ja loppuun ei toimi siinä tapauksessa. Näytepisteitä lisätään aivan liian vähän, jolloin ristikorrelaatio antaa todella huonon tuloksen, koska toisesta signaalista on vain osa mukana eikä hidasta toistoa näin ollen hyväksytty toistoksi.

Jos liikearjan loppupuolella toistot hidastuvat selvästi, mutta kuitenkin halutaan, että maksimivoimalaite rekisteröi ne toistoiksi, on silloin kehitettävä jokin luotettava ja helppo tapa signaalin skaalaamiseen. Kun vertailtavat signaalit on skaalattu saman mittaisiksi, pitäisi ristikorrelaatiovertailuun pystyä luottamaan.

Algoritmin toimintavarmuudessa on vielä paljon parantamisen varaa. Se kuitenkin toimii tällaisenaankin tarpeeksi hyvin silloin, kun kaikki tehdyt liikkeet tehdään tarpeeksi puhtaasti ja suunnilleen samalla nopeudella. Algoritmissa ei myöskään ole otettu huomioon erilaisia virhetilanteita, joita saattaa syntyä. Myös algoritmin toiminnan muuntaminen reaaliaikaiseksi tuo omia haasteita.

Tulevaisuudessa maksimivoimalaitteeseen luultavasti yhdistetään gyroskoopit, jolloin putoamiskiihtyvyyden aiheuttamat ongelmat saadaan suodatettua pois. Kun käytössä on sekä suodatettu että suodattamaton signaali, voi se tarjota mielenkiintoisia

jatkokehitysnäkymiä. Ehkä tulevaisuuden maksimivoimamittari osaa itse tulkita tehtävän liikkeen ilman käyttäjän tekemää valintaa.

Jatkossa kannattaa myös pohtia esimerkiksi gyroskooppien tuottaman datan käyttöä liikkeen tunnistamisen apuna. Sitä voisi ehkä käyttää hauiskäännön tapaisissa liikkeissä, joissa on kiertoa mukana. Periaatteessa tällä algoritmilla voidaan myös tunnistaa kulmakiihtyvyyksien perusteella laskettavien kulmien arvoja, koska yksinkertaistettuna tämän algoritmin toiminta perustuu muotojen tunnistamiseen. Algoritmi itsessään ei osaa päätellä käsiteltävän signaalin alkuperää. Tämän vuoksi riittää, kunhan käsiteltävä signaali on oikeassa muodossa, jotta algoritmi osaa käsitellä sitä.

## 6 LOPPUSANAT

Opinnäytetyön tarkoituksena oli kehittää maksimivoimalaitteeseen kuntosalilla tapahtuvat voimailuliikkeet tunnistava algoritmi. Algoritmin piti olla toimintaperiaatteeltaan sen kaltainen, että se tunnistaa riittävän hyvin tehdyt kuntoiluliikkeet, mutta ei reagoi vääränlaisiin liikkeisiin.

Algoritmin toimintavarmuuteen vaikuttavat monet seikat. Yksi tärkeimpiä asioita on se, että vertailukohteenä käytettävä referenssiliike suoritetaan mahdollisimman hyvin. Jos liikesarja suoritetaan lähes referenssinliikkeen kaltaisesti, on algoritmi kohtuullisen toimintavarma. Kuitenkin täytyy huomata, että mahdollisia virhetilanteita, toiston tunnistamatta jäämisiä tai vääriä tunnistuksia, voi syntyä monesta eri syystä.

Algoritmissa on useita asioita, joita voidaan säätää. Referenssisignaalin pituuteen voidaan vaikuttaa, liipaisukynnyksien määrää ja herkkyyttä voidaan säätää. Ristikorrelaatiovertailun hyväksymisrajaa voidaan kasvattaa tai pienentää. Eri akseleille voidaan antaa erilaisia arvoja, jolloin voidaan painottaa tiettyä liikesuuntaa.

Maan vetovoima vaikeuttaa kiihtyvyystietojen käyttämistä tämänkaltaisessa sovelluksessa. Vaikka tulevaisuudessa putoamiskiihtyvyyden vaikutus voidaan suodattaa pois, ei se ollut mahdollista tätä työtä tehtäessä.

Tuotekehitys on aina haastavaa, ja erityisen haastavaksi sen tekee se, että työskentelee yksin ja kehittää jotain lähes tyhjältä pöydältä. En kuitenkaan voi missään nimessä sanoa, että olisin ollut ilman tukea. Erityisiä haasteita aiheutti myös se, että omat ohjelmointitaitoni Matlabilla, varsinkin alussa, olivat myös aika rajalliset.

Algoritmi ei tällaisenaan ole vielä lähellekään valmis kaupalliseen tuotteeseen, mutta jatkuva kehitystyö luo mahdollisuuden saada täysin toimiva algoritmi tulevaisuudessa. Tärkeintä oli kuitenkin se, että lopuksi saatiin jotain sellaista, joka puutteineen kuitenkin toimii ja siten antaa tulevalle kehitystyölle hyvän alustan. Tämä opinnäytetyö oli myös itselleni tietynlainen näytönpaikka. Oppimisen kannalta tämän työn aikana tuli kerrattua jo opittua ja sovellettua niitä tietoja käytännössä. Oli myös paljon asioita, joista en ollut

koskaan edes kuullut. Opin myös sen, että vaikka tuotekehitys saattaa kuulostaa helpolta ja monet asiat tuntuvat teoriassa helpoilta, näin kuitenkin ei aina välttämättä ole. Maksimivoimalaite on innovatiivinen keksintö, jonka kehittämiseen ovat useat eri henkilöt antaneet panoksensa, ja olen kiitollinen siitä, että olen saanut antaa oman panokseni tämän tuotteen kehittämiseen.

## LÄHTEET

1. Kiihtyvyys. 2011. Wikipedia Vapaa tietosanakirja. Saatavissa: <http://fi.wikipedia.org/wiki/Kiihtyvyys>. Hakupäivä 22.5.2011.
2. Inkinen, Pentti – Tuohi, Jukka. 2006. Momentti 1 Insinöörfysiikka. 4. painos. Keuruu: Kustannusosakeyhtiö Otava.
3. A beginner's guide to accelerometers, Dimension Engineering LLC. Saatavissa: <http://www.dimensionengineering.com/accelerometers.htm>. Hakupäivä 22.5.2011.
4. LIS3L06AL MEMS inertial sensor: 3-axis - +/-2g/6g ultracompact linear accelerometer. 2006. Datalehti. STMicroelectronics. Saatavissa: <http://www.st.com/stonline/products/literature/od/11669/lis3l06al.pdf>. Hakupäivä 28.5.2011
5. Mäkelä, Mikko - Soininen, Lauri - Tuomola, Seppo – Öistämö, Juhani. 2001. Tekniikan kaavasto. 3. painos. Jyväskylä: Tammertekniikka.
6. Mittaustuloksen integrointi ja derivointi. 2011. Saatavissa: [http://www.helsinki.fi/~fyl\\_tlpk/luento/data-int-deri-interp.html](http://www.helsinki.fi/~fyl_tlpk/luento/data-int-deri-interp.html). Hakupäivä 22.5.2011.
7. Jerk (physics). 2011. Wikipedia The Free Encyclopedia. Saatavissa: [http://en.wikipedia.org/wiki/Jerk\\_%28physics%29](http://en.wikipedia.org/wiki/Jerk_%28physics%29). Hakupäivä 22.5.2011.
8. Ristikorrelaatio. 2010. Wikipedia Vapaa tietosanakirja. Saatavissa: <http://fi.wikipedia.org/wiki/Ristikorrelaatio>. Hakupäivä 22.5.2011.
9. Stackoverflow. 2009. Saatavissa: <http://stackoverflow.com/questions/1289415/what-is-a-good-r-value-when-comparing-2-signals-using-cross-correlation>. Hakupäivä 22.5.2011.
10. Honkanen, H. Kajaanin ammattikorkeakoulu. Suodattimet. Saatavissa: <http://gallia.kajak.fi/opmateriaalit/yleinen/honHar/ma/SUODATTIMET.pdf>. Hakupäivä 22.5.2011.

11. MathWorks. 2011. Product Documentation. Saatavissa:  
<http://www.mathworks.com/help/toolbox/signal/buttord.html>. Hakupäivä: 22.5.2011.

