



Hobuse ja ratsutaja interaktsiooni dünaamika analüüs

Bakalaureusetöö

Üliõpilane: Kirke Ilves

Üliõpilaskood: 222954YAFB

Juhendaja: Dmitri Kartofelev, Küberneetika instituut, teadur

Õppekava: Rakendusfüüsika

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Kirke Ilves

Töö vastab bakalaureusetööle esitatavatele nõuetele.

Juhendaja: Dmitri Kartofelev

Sisukord

Sisukord.....	3
1 Sissejuhatus	5
2 Teoreetilised alused.....	6
2.1 Allüürid ja tsükliline liikumine biomehaanikas.....	6
2.2 Hobune–ratsutaja kui seotud süsteem.....	6
2.3 Inertsiaalsensorid ja valitud signaalikomponendid	7
2.4 Interaktsiooni uurimine a_z ja 0_x aegridade abil	8
2.5 Interaktsiooni analüüs autokorrelatsiooni ja ristkorrelatsiooni abil	8
3 Mõõtmismeetod ja andmeanalüüs.....	10
3.1 Mõõtesubjektid ja mõõtevarustus	10
3.2 Interaktsiooni analüüs	11
3.2.1 Aegridade analüüs	11
3.2.2 Autokorrelatsioon ja ristkorrelatsioon	11
3.3 Kirjutatud tarkvara	11
4 Tulemused	12
4.1 Samm: kiirendus a_z	12
4.1.1 a_z ajasignaaliid	12
4.1.2 a_z autokorrelatsioon ja ristkorrelatsioon	14
4.2 Samm: pöördenurk ümber x-telje ϑ_x	15
4.2.1 ϑ_x ajasignaaliid	15
4.2.2 θ_x autokorrelatsioon ja ristkorrelatsioon.....	17
4.3 Galopp: vertikaalne kiirendus a_z	20
4.3.1 a_z ajasignaaliid	20
4.3.2 a_z autokorrelatsioon ja ristkorrelatsioon	20
4.4 Galopp: pöördenurk ümber x-telje ϑ_x	21
4.4.1 ϑ_x ajasignaaliid	21
4.4.2 ϑ_x autokorrelatsioon ja ristkorrelatsioon.....	23
4.5 Koondnäitajad	24
5 Kokkuvõte	26
5.1 Allüüri mõju interaktsiooni struktuurile	26

5.2	Oskustaseme mõju: kooskõla, amplituud ja stabiilsus	26
5.3	Kiirendus vs nurk: milline mõõdik töötab paremini?	27
5.4	Edasised töösuunad	27
6	Tänuavaldused	28
7	Annotatsioon	29
8	Abstract	30
9	Kasutatud kirjandus.....	31
10	Lisad	33
	Lisa 1 Pythoni moodul <code>fig0.py</code>	33
	Lisa 2 Pythoni moodul <code>fig1.py</code>	38
	Lisa 3 Pythoni moodul <code>fig2.py</code>	40
	Lisa 4 Pythoni moodul <code>fig3.py</code> ja <code>fig4.py</code>	44
	Lisa 5 Lihtlitsents	45

1 Sissejuhatus

Hobuse ja ratsutaja liikumine moodustab dünaamilise süsteemi, kus mõlema osapoole liikumismustrid mõjutavad teineteist pidevalt. Ratsaspordis kirjeldatakse sageli koostööd, harmooniat ja rütmis olemist, kuid praktikas hinnatakse seda enamasti subjektiivselt (treeneri, kohtuniku või ratsutaja tunnetuse põhjal). Samas mõjutab hobuse–ratsutaja interaktsioon otseselt nii soorituse kvaliteeti (stabiilsus, rütm, tasakaal) kui ka koormusjaotust ning potentsiaalselt hobuse heaolu [1]. Seetõttu on oluline arendada meetodeid, mis võimaldavad hobuse ja ratsutaja interaktsiooni kirjeldada kvantitatiivselt ja võrreldavalt eri alüürides ning eri ratsutajate vahel.

Viimastel aastatel on biomehaanilistes ja sporditeaduslikes uuringutes järjest enam kasutusele võetud kaasaskantavad inertsiaalsensorid, mis võimaldavad mõõta liikumist väljaspool laboritingimusi ja loomulikes treeningoludes [2]. Inertsiaalsensorite andmestikust saab analüüsida näiteks kiirendust ja orientatsiooni muutust ning hinnata nende signaalide sarnasust hobusel ja ratsutajal. Selline lähenemine on praktiline, kuna mõõtmised on suhteliselt väheinvasiivsed ja neid saab teha tavapärasel treeningkeskkonnas, kuid samal ajal nõuab see läbimõeldud signaalitöötlust ja sobivaid sensoreid, et interaktsiooni iseloomustavad tunnused oleksid usaldusväärsed ning interpretatsioon oleks üheselt mõistetav. [3]

Käesolev töö keskendub hobuse ja ratsutaja interaktsiooni katselisele mõõtmisele ning analüüsile kahel allüüril: samm ja galopp. Andmed on kogutud sama hobusega, kuid kahe erineva ratsutajaga (algaja ja oskaja), mis võimaldab vähendada hobuse individuaalsetest iseärasustest tulenevat varieeruvust ja suunata tähelepanu ratsutaja rollile interaktsioonis. Analüüsis kasutatakse eeskätt vertikaalse kiirenduse (z -telg suunaga üles) ning nurga x -telje ümber signaale, sest need komponendid peegeldavad selgelt allüürile omast rütmi, üles-alla liikumist ja keha ette–taha kallutust, mis on seotud ratsutaja stabiliseerimise ja kehahoiaku kontrolliga.

Töö aktuaalsus seisneb selles, et hobuse–ratsutaja interaktsiooni nii kvantitatiivne kui ka kvalitatiivne hinnang mõõtmistele on oluline nii treeningu tagasiside, ratsutamisoskuse hindamise kui ka biomehaaniliste mudelite arendamise seisukohalt. Hobuse puhul on tegemist efektiivse ja stabiilse liikumisega neljajalgse süsteemiga, mille dünaamikat ning kandekoormusega (ratsutajaga) kohanemist saab vaadelda kui inspiratsiooniallikat neljajalgsete robotite arendamisel. Käesolev töö on selles mõttes esimene samm suunas, kus loodusliku neljajalgse liikumise ja dünaamilise koormusega seotud interaktsiooni mõõdetud tunnuseid saaks tulevikus kasutada näiteks roboti keha stabiliseerimise, rütmi/tsüklilisuse kirjeldamise või kandekoormuse mõju modelleerimise lähteandmetena.

Eesmärgiks oli hinnata hobuse ja ratsutaja liikumise interaktsiooni sammus ja galopis, võrreldes algaja tasemega ja kogenud ratsutaja mõju sama hobuse liikumisele. Selleks töödeldakse inertsiaalsensorite andmeid, visualiseeritakse hobuse ja ratsutaja signaalide aegread ja kvantifitseeritakse interaktsiooni dünaamika ristkorrelatsiooni, autokorrelatsiooni funktsioonide aegride kaudu. Töö tulemuseks on metoodiline raamistik ja esmane võrdlev analüüs, mida saab tulevikus laiendada suuremale andmestikule ning kasutada edasisteks uuringuteks. Töös aitas tekstilist loetavust parendada tehisintellekti tööriistad nagu ChatGPT ja Microsoft CoPilot.

2 Teoreetilised alused

2.1 Allüürid ja tsükliline liikumine biomehaanikas

Hobuse liikumist käsitletakse biomehaanikas sageli kui ligikaudu perioodilist protsessi, milles liikumismuster kordub takti löikes [4]. Selline käsitlus võimaldab kirjeldada liikumist ajasignaalidena ning kasutada perioodilisusega seotud mõisteid nagu periood, sagedus ja faas [5]. Eriti oluline on see juhul, kui eesmärk on võrrelda liikumismustreid erinevates tingimustes, näiteks allüüri vahetudes või erinevate ratsutajate korral.

Samm ja galopp erinevad jalalöökide mustri ning sellest tuleneva liikumisdünaamika poolest. Samm on neljataktiline allüür, kus tugifaasid ja keha vertikaalne liikumine on üldjuhul pehmemad ning liikumistsükli korduvus võib olla tsüklis tsükklisse muutlikum kui kiiremates allüürides. Galopp on asümmeetriline kolmetaktiline allüür, kus tsükli jooksul esineb selgem rütmiline struktuur. [6]

Teoreetiliselt tähendab tsüklilisus ka seda, et ühes ja samas lõigus (ilma üleminekute, pöörete või tempo muutuseta) peaks signaalides avalduma korduv struktuur: näiteks vertikaalsuunalises kiirenduses $a_z(t)$ korduvad maksimumid ja miinimumid iga tsükli jooksul. Kui signaal on tsükliline, saab hinnata nii perioodilisust (autokorrelatsiooni abil) kui ka kahe signaali omavahelist korrelatsiooni ja ajanihet (ristkorrelatsiooni abil). Just sellepärast sobivad korrelatsioonimeetodid hobuse–ratsutaja interaktsiooni analüüsimiseks hästi.

2.2 Hobune–ratsutaja kui seotud süsteem

Ratsutamisel moodustavad hobune ja ratsutaja mehaaniliselt seotud süsteemi: hobuse kere liikumine kandub sadula ja ratsutaja keha kontaktpunktide kaudu ratsutajasse ning ratsutaja kehahoiak ja liikumine mõjutavad omakorda koormusjaotust ja liikumise stabiilsust [7]. Seda võib vaadelda kui inimese–loom interaktsiooni erijuhtu, kus kaks liikuva dünaamikaga keha on sidestatud nii geomeetriselt kui ka jõudude kaudu. Equestrian biomehaanika käsitluses rõhutatakse, et ideaalsel juhul peaks kahe sportlase liikumine olema sujuvalt seotud, kuid praktikas varieerub see seos nii hobuse, ülesande kui ratsutaja oskuste tõttu. [8]

Interaktsioon ei tähenda tingimata täiuslikku samafaasilist liikumist [10] – praktiliselt on ratsutamise eesmärk pigem stabiilne ja kontrollitud kohandumine hobuse rütmiga, et liikumine oleks sujuv ning ratsutaja ei tekitaks hobusele häirivaid lisamõjusid. Ratsutamises räägitakse stabiilsest istest ja rütmis olemisest sageli kvalitatiivselt, kuid biomehaaniliselt on need kirjeldatavad signaalide omaduste kaudu.

Kui ratsutaja liikumine on hobuse liikumisega stabiilselt seotud, siis:

- ratsutaja signaalid näitavad tsükli jooksul korduvat mustrit (perioodilisus) [10],
- ratsutaja ja hobuse signaalidel on selge seos (korrelatsioon) [9],
- ajanihe nende vahel on suhteliselt püsiv (faasinihe ei uju juhuslikult) [7].

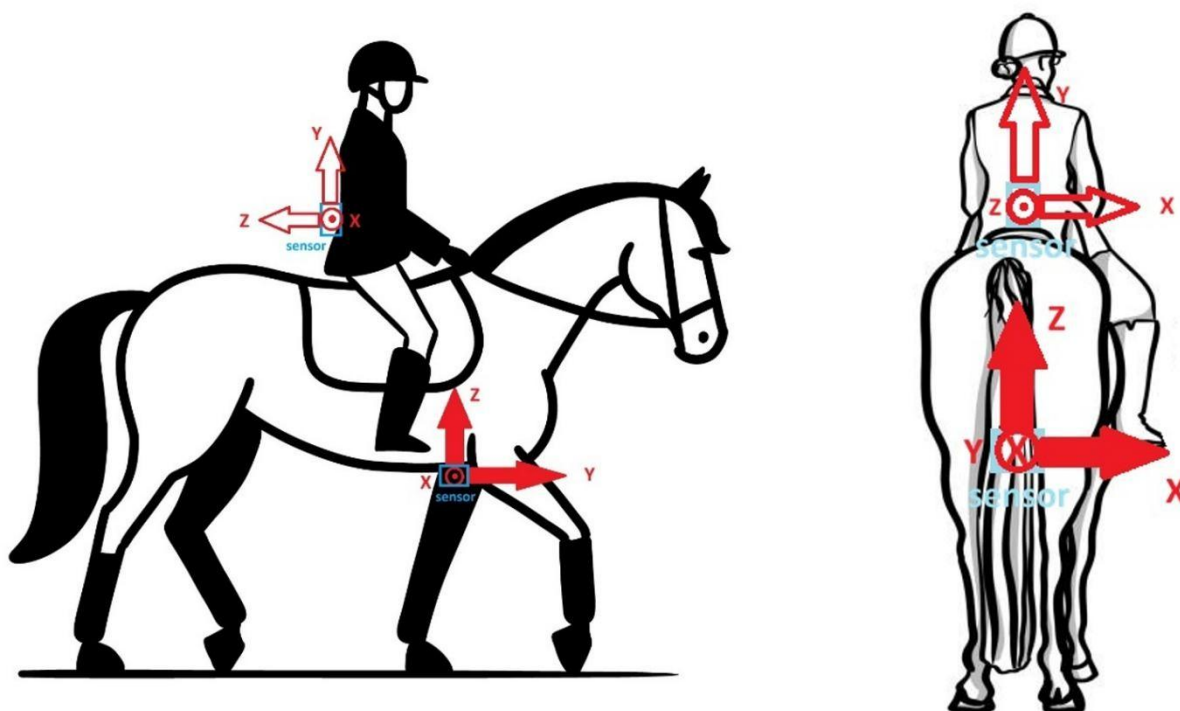
Käesolev töö ei püüa kirjeldada kõiki ratsutamise tehnilisi elemente (käed, jalad, kontakt läbi ratsmete hobuse suuga), vaid keskendub keha tasandi interaktsioonile. Seepärast sobib ratsutaja selja piirkonnast mõõdetud signaal, mis peegeldab keha üldist liikumismustrit ja kohandumist hobuse rütmiga (vt joonis 1).

2.3 Inertsiaalsensorid ja valitud signaalikomponendid

Inertsiaalsensor võimaldab mõõta liikumist kaasaskantavalt ja päris treeningtingimustes. Inertsiaalsensorite tüüpilised kanalid on:

- kiirendus (aktseleromeeter),
- nurkkiirus (güroskoop),
- nurk

Käesolevas töös on fookus kahel komponendil, milleks on vertikaalkiirendus a_z , kus z -telg on suunatud üles ning x -suunaline nurga komponent θ_x , mis kirjeldab pöörmist ümber x -telje. (vt joonis 1) Kusjuures nurk θ_x leitakse nurkkiiruse ω_x integraalist ajas.



Joonis 1. Sensorite asukohad ja koordinaatteljed kül- ja tagavaates. [13], [14]

2.4 Interaktsiooni uurimine a_z ja 0_x aegridade abil

Hobuse–ratsutaja interaktsiooni on võimalik uurida ja kirjeldada aegridade abil, võrreldes hobuse ja ratsutaja signaalide kuju, perioodilisust ning maksimumide/miinumide ajastust. Ajasignaalid annavad visuaalse ülevaate sellest:

- kui regulaarne on liikumistsükkel (kas tsüklid korduvad sarnase kujuga),
- kas signaalid on sünkroonsed või nihkes (kas maksimumid/miinumid langevad kokku),
- kui suur on tsüklitevaheline varieeruvus.

Lisaks eelmainitud aegridadele kasutame töös graafikuid, kus hobuse signaal esitatakse ratsutaja signaali funktsioonina. Taoline graafik aitab märgata telgedel kujutatud suuruste omavahelist korrelatsiooni.

Kui hobuse ja ratsutaja liikumised on perioodiliselt stabiilselt seotud, moodustub eelmainitud graafikul korduvam ja koondunum kuju. Kui seos on muutlik, tekib hajusam kuju ning sama ratsutaja väärtuse juures võib hobuse väärtus tsüklis tsükklisse erineda (trajektoor pakseneb ja ristub). Graafikute eelis on visuaalne selgus, kuid nende tõlgendus võib jääda osaliselt kvalitatiivseks. Seetõttu täiendatakse graafikuid korrelatsioonipõhiste mõõdikutega.

2.5 Interaktsiooni analüüs autokorrelatsiooni ja ristkorrelatsiooni abil

Käesolevas töös hinnatakse hobuse ja ratsutaja liikumissignaali perioodilisust ning omavahelist kooskõla korrelatsioonifunktsioonide abil. Kasutatud lähenemine põhineb diskreetajalisel signaalitöötlusel. Perioodilisuse hindamiseks arvutatakse signaali autokorrelatsioon. Pideval juhul on autokorrelatsioon:

$$(f \star f)(\tau) = \int_{-\infty}^{\infty} f(t) f(t + \tau) dt$$

kus $f(t)$ on uuritav ajasignaali aegrida, t on aja muutuja ja τ on ajanihe (aeg, mille võrra signaali ajas nihutatakse). [11]

Praktikas realiseeritakse see arvutus numbriliselt kasutades Pythoni teegi NumPy funktsiooni `numpy.correlate` [12]. Autokorrelatsioon on nulli suhtes sümmeetriline, mistõttu joonistel esitatakse sageli üks pool. Kui signaal on tugevalt perioodiline, siis autokorrelatsioonis avalduvad korduvad maksimumid ja miinumid. Nende tippude vaheline ajavahe seostub tsükli kestusega, annab visuaalse hinnangu perioodilisusele ning aitab võrrelda, kas näiteks sammus on perioodiline struktuur nõrgem/ebastabiilsem kui galopis.

Hobuse ja ratsutaja signaalide kooskõla ja ajastuse kirjeldamiseks arvutatakse ristkorrelatsioon. Antud töös on ristkorrelatsiooni integraalkuju selline:

$$(f \star g)(\tau) = \int_{-\infty}^{\infty} f(t) g(t + \tau) dt$$

kus $f(t)$, $g(t)$ on kahe mõõtmistuleme signaalide aegread (nt ratsutaja ja hobune), t on aja muutuja ja τ on ajanihe [11].

Praktikas arvutatakse ristkorrelatsioon numbriliselt ja diskreetsete andmete põhjal kasutades NumPy teegi funktsiooni `numpy.correlate` [12]. Erinevalt autokorrelatsioonist ei ole ristkorrelatsioon üldjuhul nullaja suhtes sümmeetriline. Kui hobuse ja ratsutaja signaalid on tsükliliselt seotud, siis ristkorrelatsioonis avaldub samuti võnkuv struktuur ning tekivad maksimumid, mis korduvad ligikaudu tsükli perioodiga. Ristkorrelatsiooni suurim maksimum kirjeldab korrelatsiooni tugevust, ning maksimumi asukoht τ^* näitab ajanihet, mille korral signaalide sarnasus on suurim. Seega võimaldab ristkorrelatsioon hinnata mitte ainult seda, kui sarnased signaalid on, vaid ka seda, kas üks signaal on teise suhtes ajas ees või taga (st kas parim kattuvus tekib nihkega). Korrelatsiooni integraalide võtmine on tavaline võtte. Lähenemine, kus hobuse–ratsutaja korreleeruvust kirjeldatakse ristkorrelatsiooni maksimumi ajanihke kaudu, on kasutusel ka varasemates hobuse– ratsutaja interaktsiooni uuringutes. [9]



Joonis 2. Witmotion WT9011DCL-BT50. [15]

3 Mõõtmismeetod ja andmeanalüüs

3.1 Mõõtesubjektid ja mõõtevarustus

Mõõtmised viidi läbi ühe hobusega ning osalesid kaks ratsutajat: algaja (Ratsutaja A) ja oskaja (Ratsutaja B), kes ratsutasid sama hobusega liikudes sammus ja galopis. Mõõtmistel kasutati eesti sporthobust, mära, vanusega 12 aastat ja turjakõrgusega 165 cm. Hobune oli treenituselt harrastushobune ja treenis ligikaudu 5 korda nädalas. Mõõtmiste ajal kasutati tavalist treeningvarustust ning hobune oli rauatamata. Mõõtmiste ajal ei täheldatud hobusel märkimisväärsed liikumishäireid.

Ratsutaja A (algaja) oli vanuses 11 aastat, pikkusega 160 cm ja kehakaaluga 38 kg ning varasem ratsutamiskogemus puudus. Ratsutaja B (oskaja) oli vanuses 23, pikkusega 185 cm ja kehakaaluga 68 kg ning ratsutamiskogemusega ligikaudu 16 aastat. Mõlema ratsutaja puhul kasutati sama mõõteprotokolli ja sama sensorite paigutust.

Mõõtmised viidi läbi maneežis, sirgel lõigul, paremat pidi liikudes, et vähendada trajektoorist ja suunamuutustest tulenevaid lisamõjusid. Analüüsiks valiti lõigud, kus liikumine oli stabiilne (ilma üleminekute ja selgete rütmihäireteta), et tsüklilisus oleks korrelatsioonimeetoditega hinnatav.

Liikumise mõõtmiseks kasutati kahte inertsiaalsensorit nimega Witmotion WT9011DCL-BT50 (vt joonis 2), mis salvestasid kiirendust a_z ning nurkkiirust ω_x . Proovivõtusagedus oli 200 Hz.

Sensorid paigutati järgmiselt:

- hobuse sensor kinnitati sadulavöö külge keskele alla (vt joonis 1),
- ratsutaja sensor kinnitati seljale (vt joonis 1).

Koordinaatteljed defineeriti nii, et z -telg on suunatud üles, y -telg ette (liikumissuunas) ja x -telg paremale (vt joonis 1). Kuna hobuse ja ratsutaja andurite paigaldusasend erines, viidi andmetöötluses teljed ühtsesse koordinaatraamistikku, et võrreldavad komponendid kirjeldaksid sama füüsilist suunda mõlemas signaalis. Nii tagatakse, et arvutatud korrelatsiooni näitajad sõltuvad liikumisest, mitte anduri orientatsiooni erinevusest.

3.2 Interaktsiooni analüüs

3.2.1 Aegridade analüüs

Hobuse ja ratsutaja koostoime visuaalseks hindamiseks loome mõõdetud suuruste ajasignaalide aegridade graafikute võrdlust ning moodustame hobuse–ratsutaja graafikud, kus hobuse signaal esitati ratsutaja signaali funktsioonina. Graafikud koostati eraldi vertikaalkiirenduse ja pöördenurga signaalide jaoks.

Graafikud annavad kvalitatiivse ja visuaalselt selge ülevaate sellest, kui stabiilselt on hobuse ja ratsutaja signaalid omavahel korreleerunud. Graafikute koostamiseks ühtlustati hobuse ja ratsutaja signaalide pikkused sama analüüsilõigu ulatuses.

3.2.2 Autokorrelatsioon ja ristkorrelatsioon

Interaktsiooni kvantitatiivseks kirjeldamiseks arvutati iga analüüsilõigu kohta autokorrelatsioon ning hobuse ja ratsutaja signaalide vaheline ristkorrelatsioon. Autokorrelatsiooni kasutati signaali perioodilise korduvuse hindamiseks ning ristkorrelatsiooni hobuse ja ratsutaja signaalide omavahelise korrelatsiooni ja ajastuse kirjeldamiseks. Tulemuste kokkuvõtlikuks võrdlemiseks eraldati korrelatsioonianalüüsist koondnäitajad, mille hulgas on ka ristkorrelatsiooni funktsioonide korrelatsiooni koefitsiendid (vt tabel 1).

3.3 Kirjutatud tarkvara

Andmetöötlus ja graafikute genereerimine teostati kasutades programmeerimiskeelt Python. Analüüsivoog on jaotatud eraldi moodulitesse, mis loevad sama sisendandmestikku ja moodustavad joonised ühtses formaadis:

- `fig0.py` – toorandmete lugemine, sensori alusel jagamine, ajasünkroniseerimine, vahefailide (.npy) loomine ning abifunktsioonid,
- `fig1.py` – nurga ajasignaalid ja “nurk vs nurk” graafikud,
- `fig2.py` – nurga autokorrelatsioon ja ristkorrelatsioon,
- `fig3.py` – kiirenduse ajasignaalid ja “kiirendus vs kiirendus” graafikud,
- `fig4.py` – kiirenduse autokorrelatsioon ja ristkorrelatsioon.

Moodulite `fig0.py`, `fig1.py` ja `fig2.py` sisu on lisatud selle yöö Lisad sektsiooni (vt Lisa 1, 2, 3).

4 Tulemused

Tulemused on rühmitatud allüüri ja mõõtesuuruse lõikes ning esitatud graafikutel (a_z , θ_x aegread, graafikud, kus on hobuse andmed esitatud ratsutaja anmete funktsioonina ning auto- ja ristkorrelatsioonidena). Kuna mõõtmised tehti sama hobusega, tõlgendatakse erinevusi eeskätt ratsutaja liikumisstrateegia ja hobuse rütmiga kohandumise erinevusena.

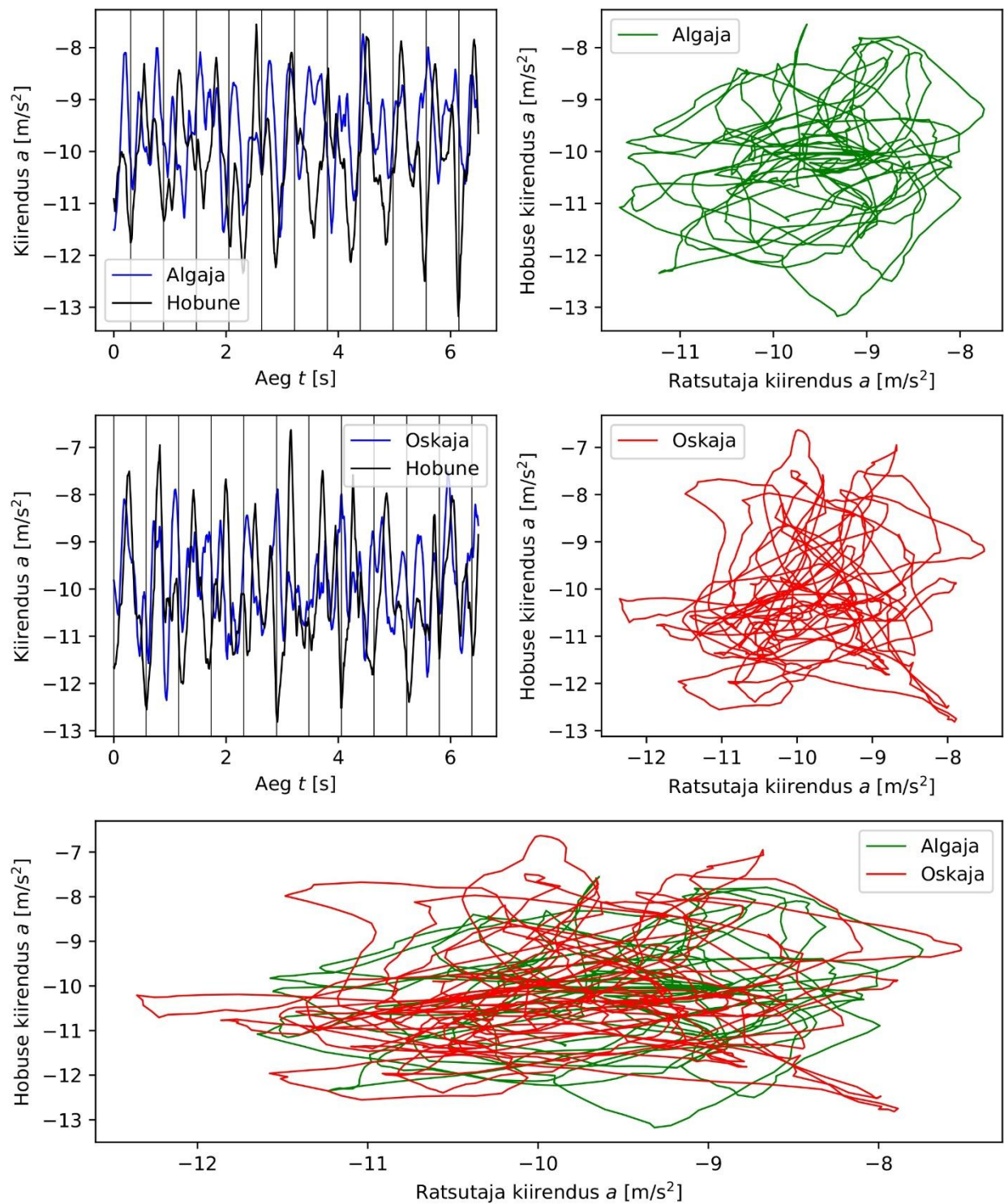
4.1 Samm: kiirendus a_z

4.1.1 a_z ajasignaalid

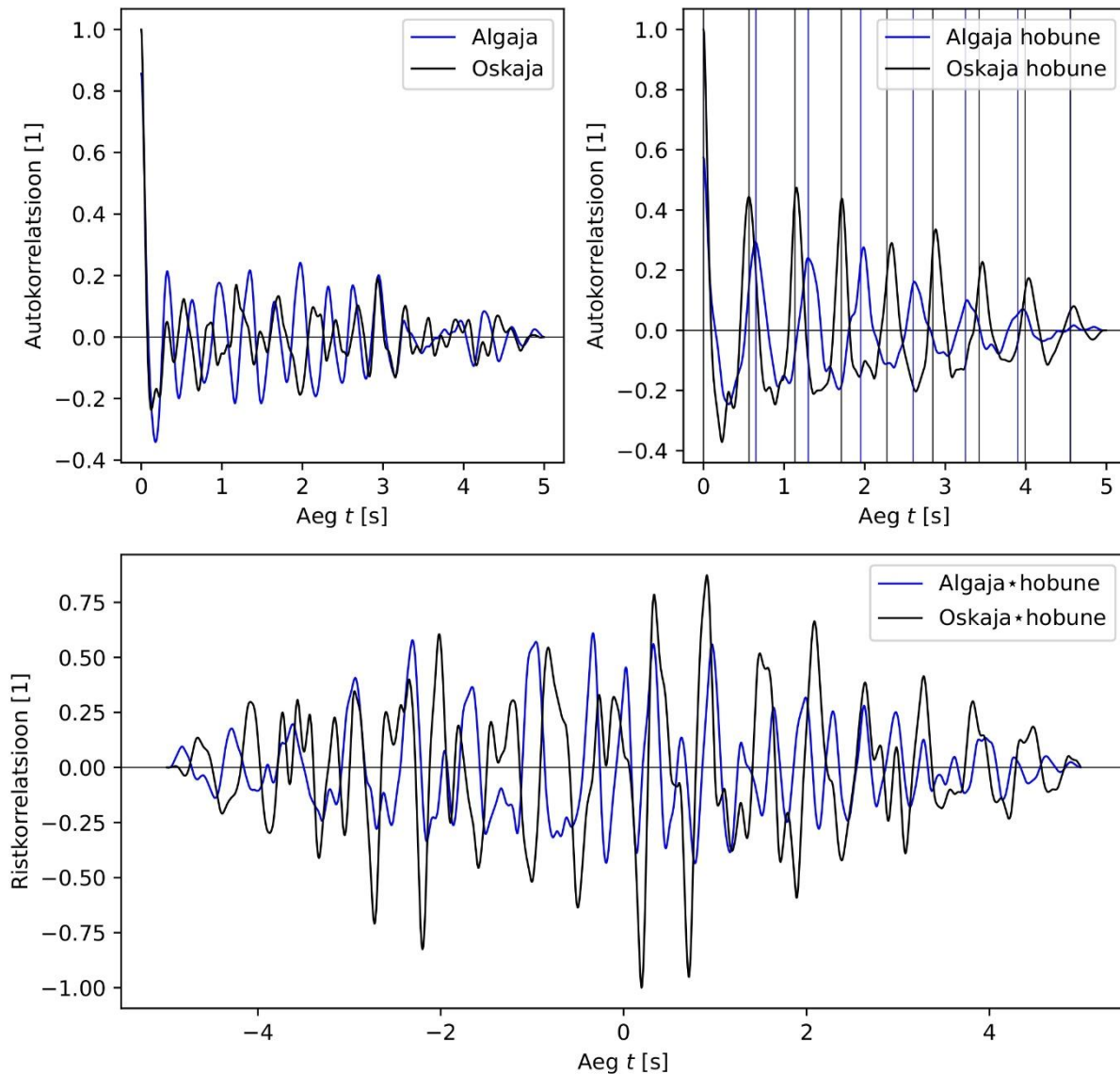
Joonisel 3 ülemisel vasakul paneelil on algaja ratsutaja ajasignaal koos hobuse signaaliga. Mõlemad signaalid võnguvad samas ajavahemikus ning on näha perioodilist struktuuri. Ratsutaja signaal järgib hobuse rütmiga kohati sarnaselt, kuid maksimumide ja miinimumide täpne ajastus ning amplituud varieerub perioodide lõikes märgatavalt, mis viitab sammuhulga halvale korrelatsioonile.

Keskmisel vasakul paneelil on oskaja ratsutaja ajasignaal koos hobuse signaaliga. Ka siin on perioodiline muster vähesel määral olemas, kuid ratsutaja signaali kuju on mitmes lõigus hobuse signaaliga ühtlasemalt seotud: ekstreemumid esinevad regulaarsemalt ning signaali üldine rütm püsib stabiilsem. Samas on sammuhulga ajasignaalides näha, et ka oskaja puhul pole visuaalselt head korreleeruvust.

Sammuhulga a_z signaalides on perioodiline muster olemas, kuid perioodide kuju ja ajastus varieeruvad lõigu jooksul ning hobuse–ratsutaja graafikud on laialivalguvad. Seetõttu ei moodustu sammus vertikaalkiirenduse põhjal üheselt eristatavat korrelatsiooni ning oskustaseme erinevust on selles mõõdikus keeruline üheselt välja tuua.



Joonis 3. Kiirenduse a_z ajasignaali ja trajektoori sammus.



Joonis 4. Kiirenduse a_z aegridade autokorrelatsioon ja ristkorrelatsioon sammus.

4.1.2 a_z autokorrelatsioon ja ristkorrelatsioon

Joonisel 4 ülemisel vasakul on kujutatud ratsutaja vertikaalkiirenduse a_z autokorrelatsioon algaja ja oskaja ratsutaja korral.

Algaja (sinine) ja oskaja (must) ratsutaja autokorrelatsioon algab väärtuselt 1 ajahetkel $t = 0$ ning langeb seejärel järsult juba esimeste kümnendike sekundi jooksul. Edasi on näha väikese amplituudiga võnkumist nulli ümber: korduvad tipud küll esinevad, kuid nende kõrgus on madal ja tippude kuju ei püsi löigu lõikes ühtlane. Algaja kõver on rahutum — võnkumisi ja väiksemaid kõrvaltippe on rohkem — samal ajal kui oskaja kõver on siledam. Mõlemal juhul sumrub autokorrelatsioon mõne sekundi jooksul nulli lähedale, mis viitab, et ratsutaja a_z perioodilisus on sammus nõrgalt väljendunud.

Hobuse a_z autokorrelatsioonis on perioodiline struktuur selgem kui ratsutajal: korduvad maksimumid ja miinimumid on paremini eristatavad ning nende vaheline ajavahe püsib ühtlasemalt. Samal ajal väheneb tippude amplituud lag'i kasvades ning umbes 4–5 sekundi piirkonnas on kõver juba nulli lähedal. Algaja ja oskaja sõidu hobuse autokorrelatsioonid on kuju poolest sarnased, mis on ootuspärane, kuna mõõdeti sama hobust - erinevused avalduvad peamiselt tippude amplituudis ja hääbumise kiiruses.

Ristkorrelatsioonis on samuti võnkuv struktuur, kuid sammus ei eristu üks selgelt domineeriv maksimum ühe kindla ajanihke juures. Selle asemel esineb mitu võrreldava suurusjärguga tippu nii negatiivsetel kui positiivsetel ajahetkedel, mis näitab, et signaalide parim kattuvus ei püsi ühe püsiva nihke ümber. Oskaja– hobune kõveral on mitmes piirkonnas maksimumid kõrgemad ja kontrastsemad kui algaja– hobune kõveral, mis viitab paremale interaktsioonile oskaja ratsutaja korral, kuid sammu puhul jääb ajastussuhe siiski muutlikuks. Lõigu 4.1 esitatud tulemus iseloomustab juhuslikkuse domineerimist perioodilisuse üle.

4.2 Samm: pöördenuk ümber x -telje θ_x

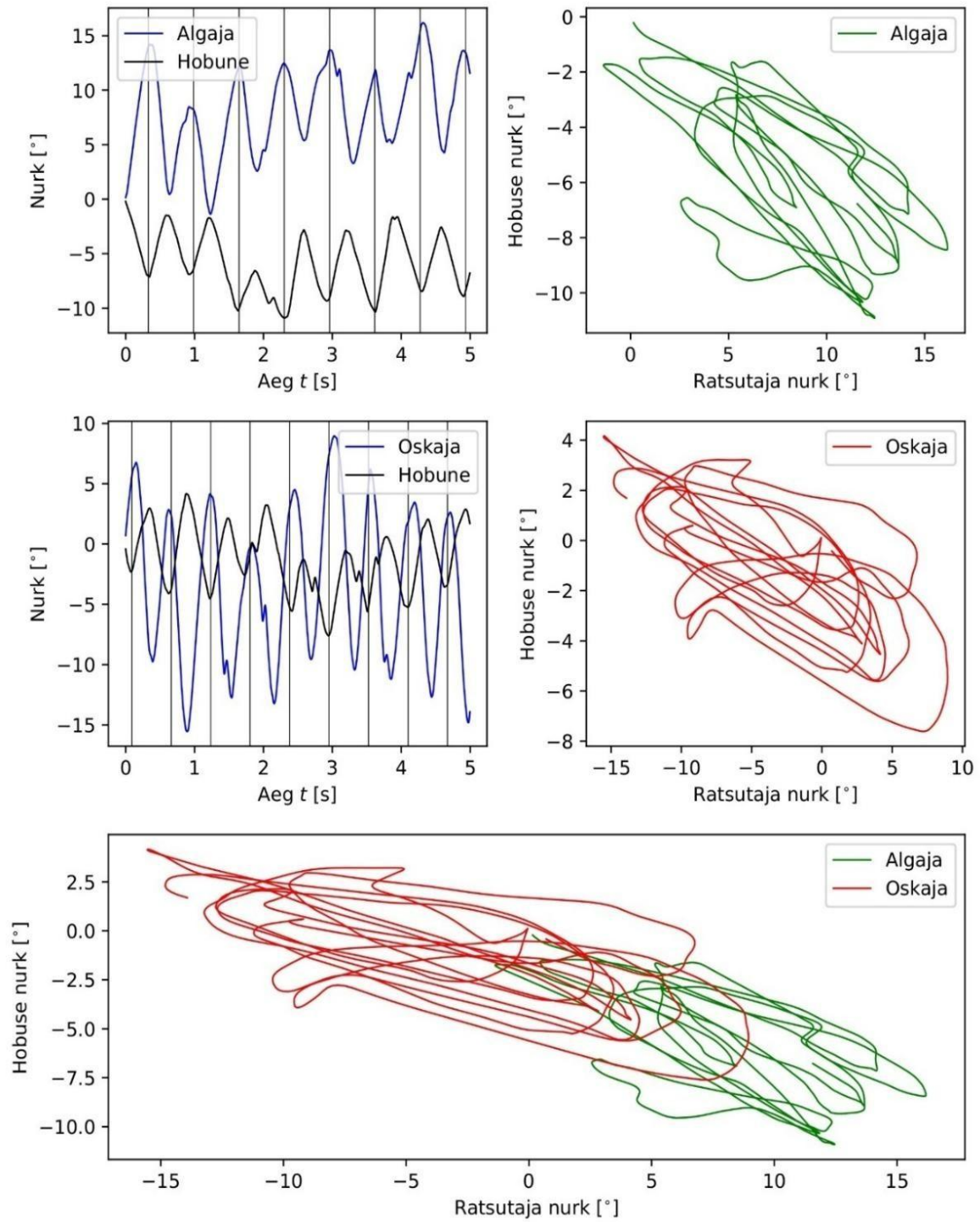
4.2.1 θ_x ajasignaalid

Joonisel 5 on esitatud hobuse ja ratsutaja pöördenuk ümber x -telje θ_x sammus algaja ja oskaja ratsutaja korral ning vastavad “hobuse nurk vs ratsutaja nurk” graafikud.

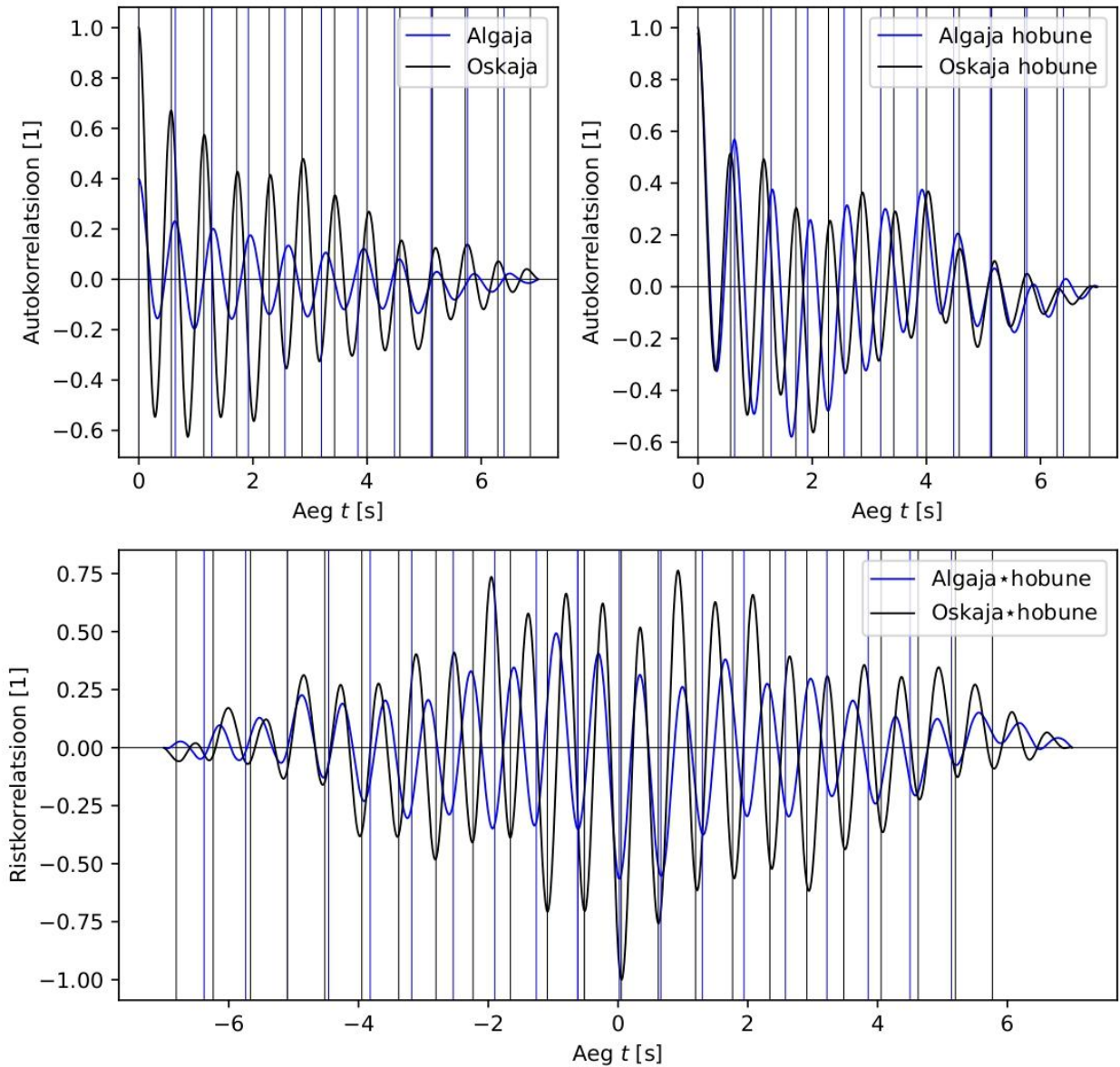
Ülemisel vasakul paneelil (algaja) võngub ratsutaja nurk perioodiliselt ja püsib valdavalt positiivses vahemikus (ligikaudu $0 \dots 15^\circ$), samal ajal kui hobuse nurk paikneb valdavalt negatiivses vahemikus (ligikaudu $0 \dots -10^\circ$). Signaalides on perioodiline muster olemas, maksimumide ja miinimumide kõrgus varieerub lõigu jooksul, mistõttu amplituud ja kuju ei ole periooditi ühesugused.

Keskmisel vasakul paneelil (oskaja) on ratsutaja θ_x amplituud suurem ning signaal ulatub ligikaudu vahemikku $-15 \dots +10^\circ$, samal ajal kui hobuse θ_x püsib kitsamas vahemikus. Ka oskaja puhul on perioodilisus selgelt olemas, kuid sammu korral on näha muutlikkust (amplituud ja kuju kõiguvad).

Parempoolsetel paneelidel esitatud “hobuse nurk vs ratsutaja nurk” graafikud näitavad, et θ_x põhjal on hobuse ja ratsutaja vahel negatiivne korrelatsioon: graafikul joon on valdavalt kaldu allapoole ehk suurema ratsutaja nurga juures on hobuse nurk väiksem (rohkem negatiivne) ja vastupidi. See tähendab, et hobuse ja ratsutaja θ_x muutused on sammus sageli vastassuunalised. Algaja trajektoor on laiem ja ebakorrapärasem, oskaja trajektoor on mitmes piirkonnas koondunud.



Joonis 5. Nurga θ_x ajasignaali sammus.



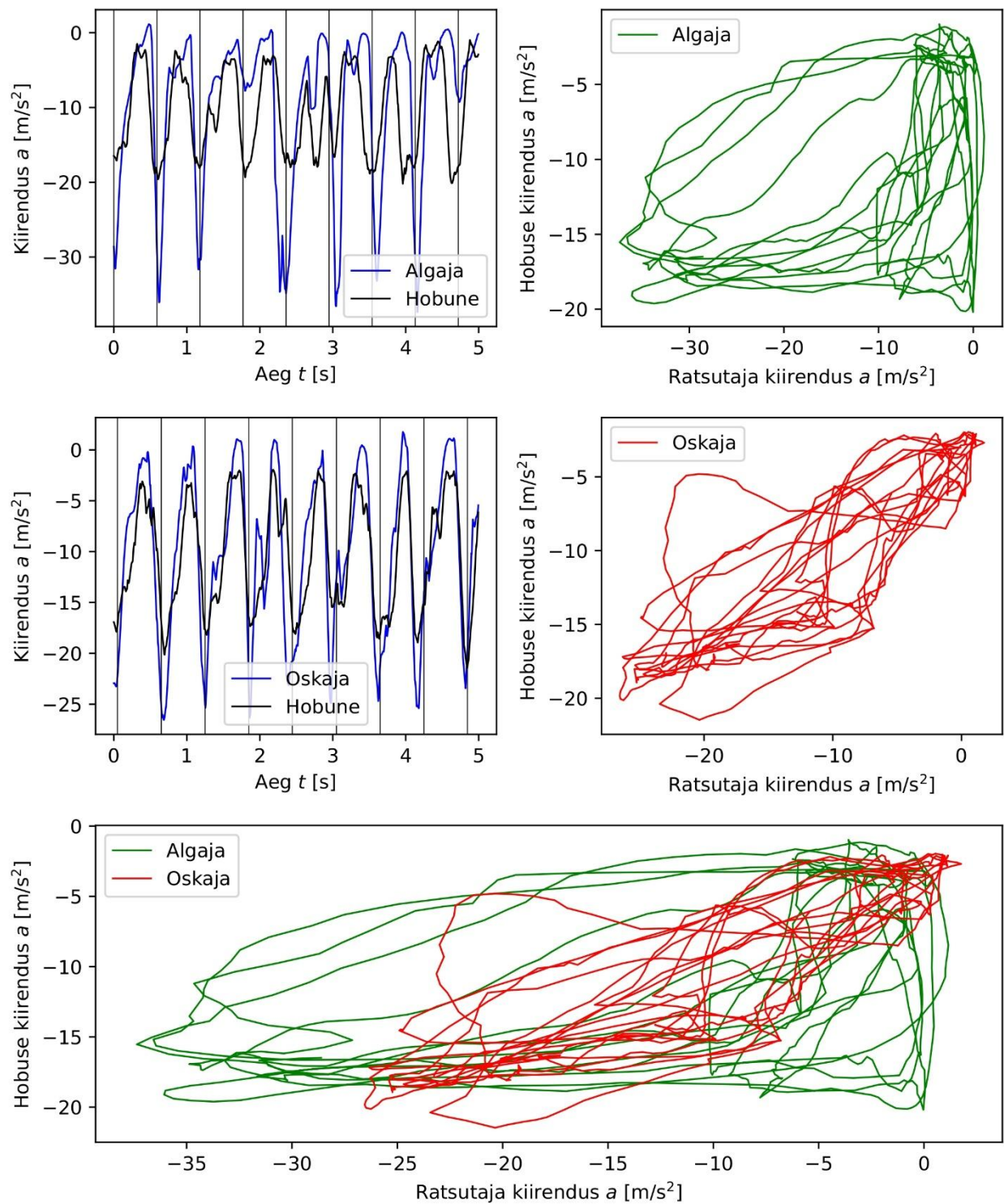
Joonis 6. Nurga θ_x aegridade autokorrelatsioon ja ristkorrelatsioon sammus.

4.2.2 θ_x autokorrelatsioon ja ristkorrelatsioon

Joonisel 6 ülemisel vasakul on kujutatud ratsutaja θ_x autokorrelatsioon algaja (sinine) ja oskaja (must) ratsutaja korral. Oskaja autokorrelatsioonis on korduvad maksimumid ja miinimumid selgemalt eristatavad ning võnkumise amplituud on suurem, mis viitab ühtlasemale korduvusele ja tugevamale perioodilisele komponendile ratsutaja orientatsioonimuutuses. Algaja autokorrelatsioonis on amplituud väiksem ning võnkumine hääbub kiiremini, mis viitab suuremale perioodivahelisele varieeruvusele.

Joonise ülemisel paremal on esitatud hobuse θ_x autokorrelatsioon sammus. Mõlema sõidu korral on hobuse autokorrelatsioonidel selge perioodiline kuju ning tipud ja miinimumid paiknevad sarnastel ajanihetel. Amplituudis ja summutumiskiiruses esineb siiski mõningaid erinevusi.

Joonise alumisel paneelil on ristkorrelatsioon hobuse ja ratsutaja θ_x signaalide vahel. Mõlema paari ristkorrelatsioon on selgelt võnkuv ja perioodiline, mis näitab, et hobuse ja ratsutaja θ_x signaalidel on sammas olemas süsteemne seos. Silmatorkavalt ei ole ristkorrelatsioon $t = 0$ juures domineerivalt positiivne; $t = 0$ läheduses esineb tugev negatiivne äärmus, mis viitab, et hobuse ja ratsutaja θ_x muutused on selles lõigus sageli vastassuunalised (anti-korrelatsioon). Ristkorrelatsiooni funktsiooni oskaja* θ_x hobune kõveral on ristkorrelatsiooni maksimumide ja miinimumide muster üldiselt kontrastsem ja amplituud suurem, mis viitab paremale korreleeruvusele oskaja ratsutaja korral. Lisaks esinevad suuremad maksimumid/miinimumid mitmel juhul nihkes null-lagist, mis tähendab, et parim kattuvus saavutatakse teatud ajanihkega, mitte täpselt samal ajahetkel.



Joonis 7. Kiirenduse a_z ajasignaalid galopis.

4.3 Galopp: vertikaalne kiirendus a_z

4.3.1 a_z ajasignaaliid

Joonisel 7 on esitatud hobuse ja ratsutaja vertikaalne kiirendus a_z galopis ning vastavad graafikud. Ajasignaaliidest on näha, et valitud lõigus on galopi liikumine selgelt perioodiline: 5 sekundi jooksul kordub mitmeid sarnase kujuga tsükleid nii hobuse kui ratsutaja signaaliides. Hobuse signaaliid üldkuju on kahe ratsutaja korral sarnane (maksimumid ja miinimumid korduvad samas rütmilises struktuuris), kuid amplituud ja detailid erinevad mõnevõrra.

Ratsutaja signaaliis eristub algaja ja oskaja vahel selge amplituudi erinevus. Algaja ratsutaja a_z minimaalsed väärtused ulatuvad ligikaudu -35 m/s^2 suurusjärku, samas kui oskajal jäävad minimaalsed väärtused ligikaudu $-25 \dots -27 \text{ m/s}^2$ piirkonda. Algaja signaaliis esineb mitmes tsüklis teravam sukeldumine (järsemad ja sügavamad miinimumid), samal ajal kui oskaja signaal on sujuvama kujuga ning sarnaneb hobuse rütmilisele struktuurile ühtlasemalt.

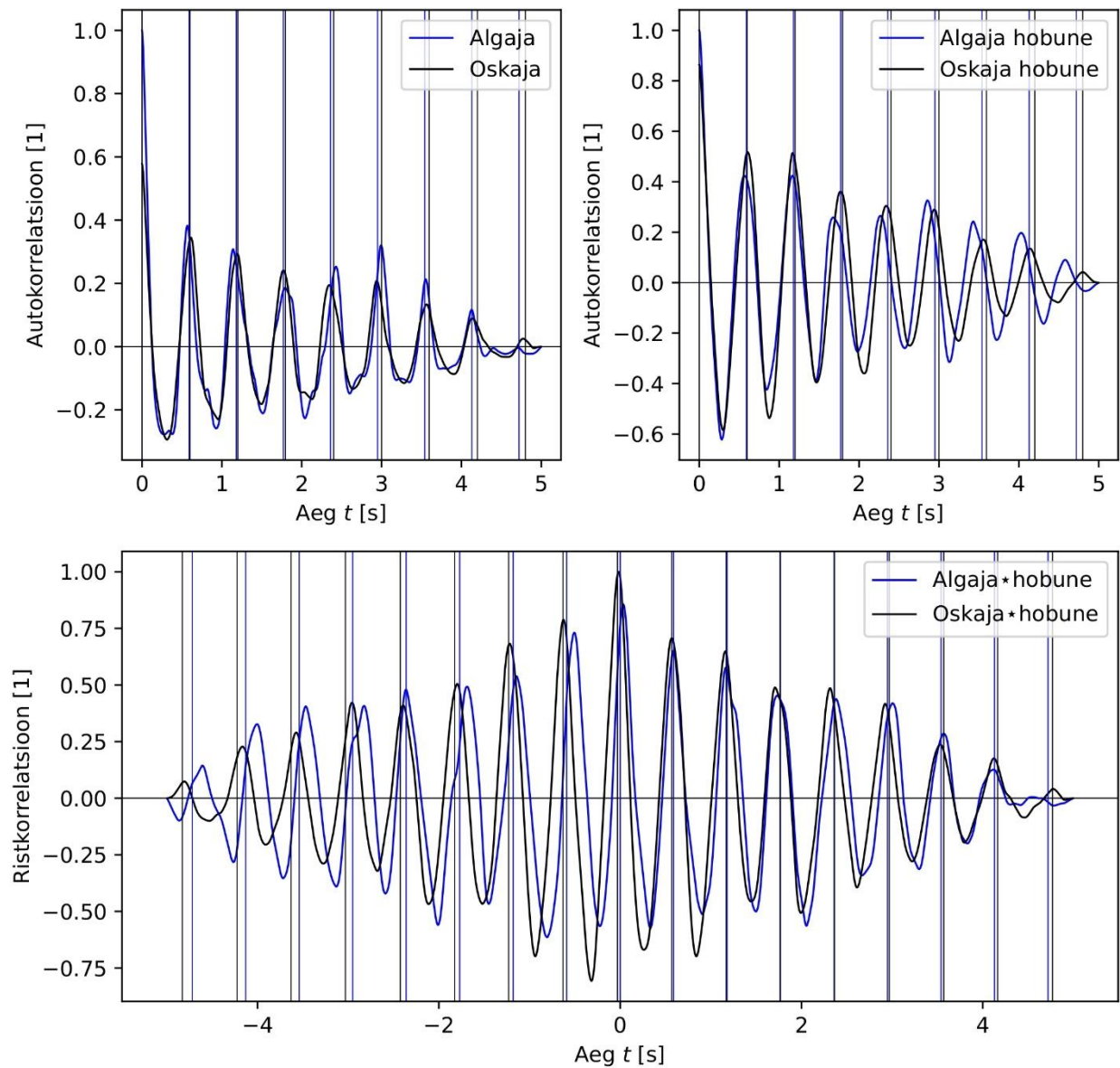
Graafikud kinnitavad sama erinevust teise vaatenurga alt. Algaja trajektoor katab suurema ala ning moodustab laiemad ja ebaühtlasemad silmused, oskaja trajektoor on koondunum ja paikneb kitsamas piirkonnas. Alumisel koondjoonisel on näha, et oskaja trajektoor paikneb suures osas algaja graafiku sees ning sisaldab vähem äärmuslikke ratsutaja kiirendusväärtusi.

4.3.2 a_z autokorrelatsioon ja ristkorrelatsioon

Joonisel 8 ülemisel vasakul graafikul on esitatud ratsutaja vertikaalkiirenduse a_z autokorrelatsioon algaja ja oskaja ratsutaja korral.

Joonise ülemisel paremal on näidatud hobuse a_z autokorrelatsioon algaja ja oskaja sõidu korral. Ka hobuse autokorrelatsioonides esineb selge summutatud perioodiline struktuur, kuid võrreldes ratsutajaga on alguses näha tugevamat võnkumist (suuremad positiivsed ja negatiivsed ekstreemid).

Joonise alumisel paneelil on ristkorrelatsioon hobuse ja ratsutaja a_z vahel (algaja**hobune* ja oskaja**hobune*). Mõlema paari ristkorrelatsioonil on perioodiline struktuur, mis tähendab, et hobuse ja ratsutaja signaaliid jagavad sama põhisagedust (galopi periood). Kõige olulisem tulemus on see, et ristkorrelatsiooni kõrgeim maksimum asub ligikaudu $t = 0$ lähedal ehk parim kattuvus saavutatakse peaaegu ilma ajalise nihketa, mis tähendab, et süsteem on sünkroniseeritud. Lisaks on oskaja**hobune* kõvera maksimum oluliselt kõrgem kui algaja**hobune* kõveral.

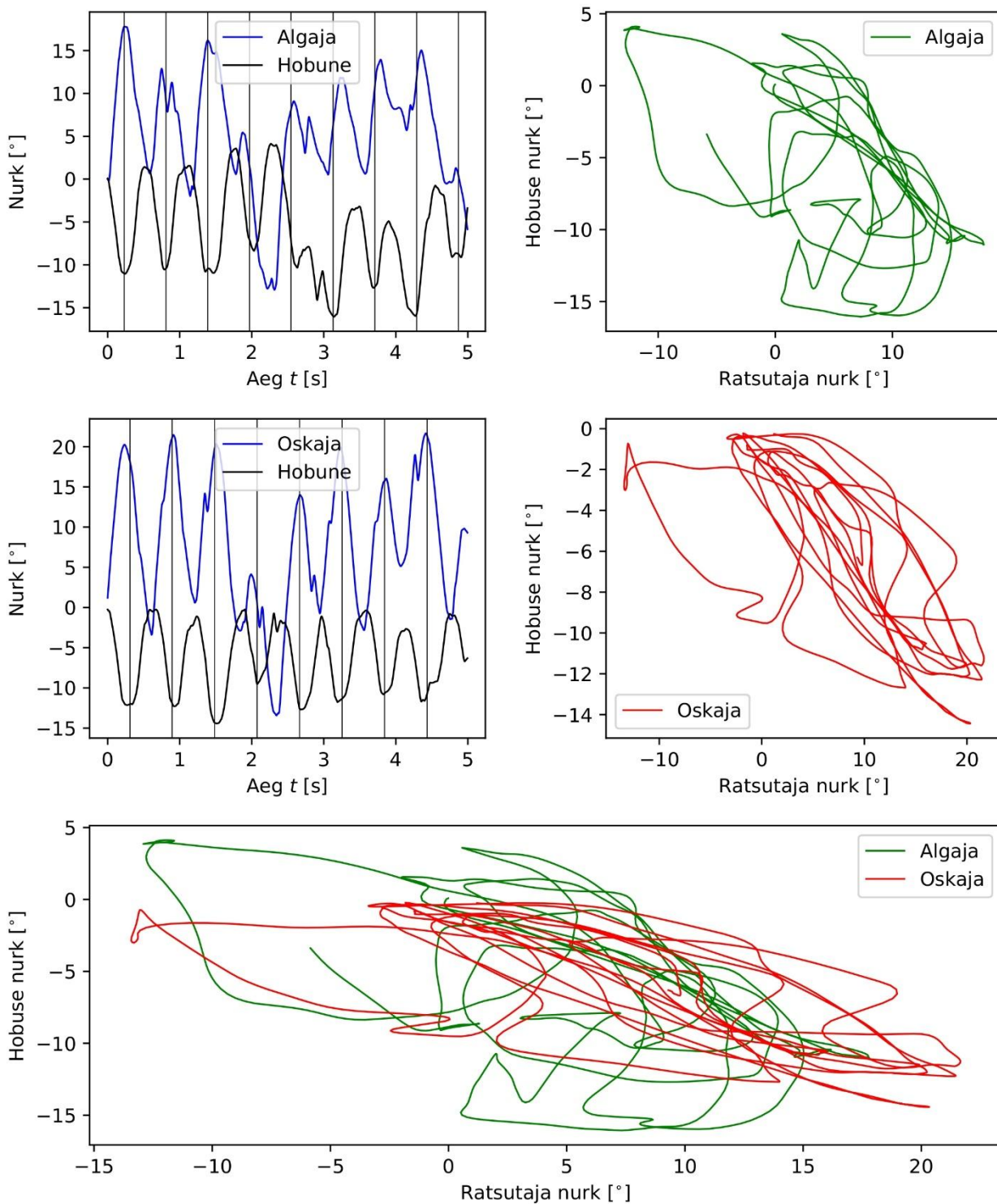


Joonis 8. Kiirenduse a_z aegridade autokorrelatsioon ja ristkorrelatsioon galopis.

4.4 Galopp: pöördenurk ümber x -telje ϑ_x

4.4.1 ϑ_x ajasignaali

Joonisel 9 on esitatud hobuse ja ratsutaja pöördenurk ümber x -telje ϑ_x parema jalaga galopis algaja ja oskaja ratsutaja korral. Ajasignaali (vasakul) on mõlema ratsutaja puhul näha selget tsüklilist mustrit: hobuse nurk võngub regulaarselt ning ratsutaja nurk kordab samuti rütmilist struktuuri, kuid suurema amplituudiga. Algaja korral varieerub ratsutaja nurk ebastabiilselt: tsüklite amplituud ja kuju muutuvad rohkem lõigu jooksul ning signaalis esineb suuremaid kõikumisi ümber keskmise taseme.



Joonis 9. Nurga θ_x ajasignaali ja trajektooriid galopis.

Oskaja korral on ratsutaja nurga võnkumine ühtlasem ja perioodilisem, kuigi amplituud on endiselt suurem kui hobusel.

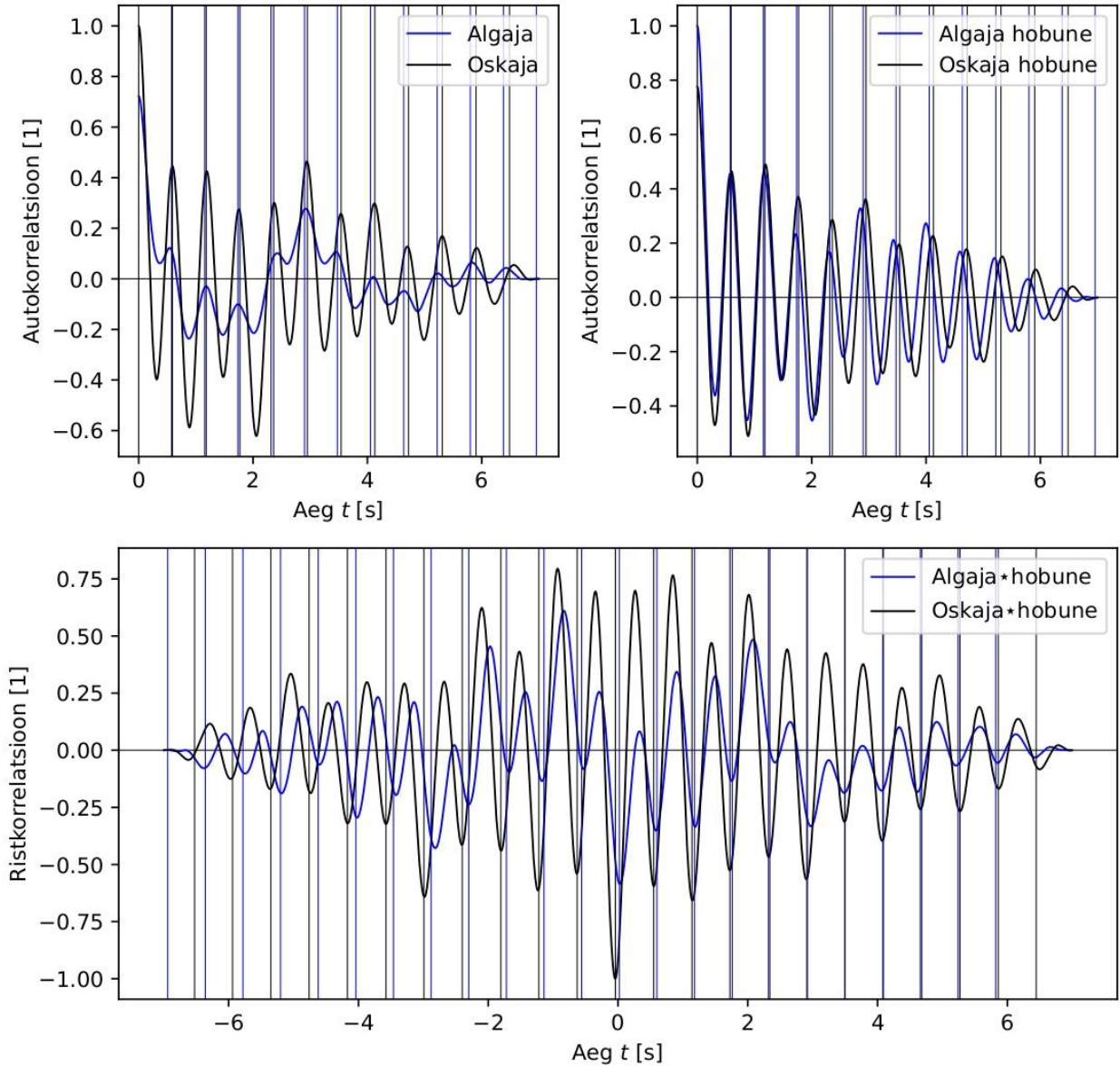
Graafikud, kus hobuse nurk on kujutatud ratsutaja nurga funktsioonina, näitavad algaja ja oskaja vahel erinevust interaktsiooni stabiilsuses. Algaja trajektoor on hajus ja katab suurema ala ning moodustab ebaühtlaseid ja ristuvaid silmuseid. Oskaja trajektoor on koondunud ja korrapärasema kujuga ning paikneb kitsamas piirkonnas.

4.4.2 θ_x autokorrelatsioon ja ristkorrelatsioon

Joonisel 10 ülemisel vasakul on kujutatud ratsutaja θ_x autokorrelatsioon algaja ja oskaja ratsutaja puhul. Mõlema ratsutaja autokorrelatsioonides on näha selge perioodiline struktuur, kuid oskaja korral on korduvad maksimumid ja miinimumid kontrastsemad ning amplituud püsib ajanihete kasvades kauem suurem. Algaja autokorrelatsioon summutub kiiremini ja kuju on ebaühtlasem, mis viitab suuremale perioodidevahelisele varieeruvusele.

Joonise ülemisel paremal on hobuse θ_x autokorrelatsioon algaja ja oskaja sõidu korral. Mõlemal juhul on perioodiline muster selgelt nähtav ning summutatud võnkumise kuju on sarnane.

Joonise alumisel paneelil on ristkorrelatsioon hobuse ja ratsutaja θ_x signaalide vahel. Ristkorrelatsioonikõverad on perioodilised ning mõlema ratsutaja korral esinevad korduvad tipud, mis näitab süsteemset seost hobuse ja ratsutaja nurkliikumise vahel galopi tsüklis. Aja $t = 0$ läheduses on ristkorrelatsioon negatiivne ehk hobuse ja ratsutaja θ_x muutused on selles mõõdikus sageli vastassuunalised. Oskaja**hobune* kõveral on maksimumid ja miinimumid üldiselt kontrastsemad ja suurema amplituudiga, mis viitab paremale korrelatsioonile, algaja**hobune* kõveral on amplituud väiksem ja muster ebaühtlasem. Lisaks ei paikne suurimad äärmused alati täpselt samal ajahetkel, mis viitab, et parim kattuvus võib tekkida väikese ajanihkega.



Joonis 10. Nurga θ_x aegridade autokorrelatsioon ja ristkorrelatsioon galopis.

4.5 Koondnäitajad

Tabel 1 koondab hobuse ja ratsutaja interaktsiooni näitajad sammus ja galopis vertikaalkiirenduse a_z ning pöördenurgaga θ_x . Vertikaalkiirenduse korral on sammus korrelatsioon väga väike (algaja $r = 0.20$, oskaja $r = 0.02$), mis tähendab, et a_z signaalide omavaheline seos ei ole selles allüüris stabiilselt väljendunud ning oskustaseme erinevus ei tule a_z järgi sammus selgelt esile. Mõlema ratsutaja puhul on faasiinihe sama suurusjärgus, kuid kuna korrelatsioon on nõrk, tuleb seda ajastust käsitleda pigem orientiirina kui tugeva järeldusena. Sammu a_z põhjal ei ole tabelis toodud ka karakteristlikku perioodi, mis viitab, et perioodilisuse hinnangut ei olnud selles signaalis usaldusväärselt eristatav.

Tabel 1. Hobuse–ratsutaja interaktsiooni koondnäitajad sammus ja galopis vertikaalkiirenduse a_z ning pöördenurga ümber x -telje θ_x põhjal (algaja vs oskaja ratsutaja).

		Korrelatsiooni koefitsient		Faasinihe (ratsutaja tuleb enne) (s)		Karakteristlik periood (s)	
		algaja	oskaja	algaja	oskaja	algaja	oskaja
a_z	samm	0,20	0,02	0,24	0,24	-	-
	galopp	0,43	0,78	0,00	0,03	0,59	0,60
θ_x	samm	-0,71	-0,69	-0,02	-0,05	0,64	0,57
	galopp	-0,49	-0,76	-0,02	0,04	0,58	0,59

Galopis on a_z tulemused selgemad: korrelatsiooni koefitsient suureneb märgatavalt ning oskaja ratsutaja puhul on seos hobusega tugev (algaja $r = 0,43$, oskaja $r = 0,78$). See tähendab, et galopis on vertikaalkiirenduse signaalide kooskõla paremini väljendunud ning oskaja ratsutaja liikumine on hobuse rütmiga stabiilsemalt seotud. Faasinihe on galopis väga väike (0,00–0,03 s), mis näitab, et parim kattuvus saavutatakse peaaegu ilma ajaniheta (sünkroonis). Galopi karakteristlik periood on mõlemal ratsutajal praktiliselt sama ($\approx 0,59$ –0,60 s), mis on kooskõlas selgema perioodilise struktuuriga galopis.

Pöördenurga θ_x põhjal on seos nii sammus kui galopis negatiivne, st hobuse ja ratsutaja θ_x muutused on valdavalt vastassuunalised. Sammu korral on korrelatsioon tugevasti negatiivne ja sarnane mõlemal ratsutajal (algaja $r = -0,71$, oskaja $r = -0,69$), mis viitab, et θ_x mõõdikus on sammus olemas selge ja süsteemne vastassuunaline sidestus sõltumata oskustasemest. Faasinihe ratsutaja ja hobuse aegridade vahel on sammus väike ja negatiivne ($-0,02$ ja $-0,05$ s), mis tähendab, et parim kattuvus tekib väikese ajastusega, kus ratsutaja signaal jääb hobuse signaalist natuke maha.

Galopis on θ_x korrelatsioon samuti negatiivne, kuid oskaja ratsutaja korral tugevam (algaja $r = -0,49$, oskaja $r = -0,76$), mis näitab, et galopis eristab θ_x mõõdik oskustaset paremini kui sammus. Faasinihe on galopis nulli lähedal ($-0,02$ s ja $0,04$ s), mis tähendab, et parim kattuvus saavutatakse sisuliselt samaaegselt või väga väikese nihkega (märgi erinevus näitab vaid, kas ratsutaja on minimaalselt ees või maas). Galopi karakteristlik periood on θ_x puhul mõlemal ratsutajal praktiliselt sama ($\approx 0,58$ –0,59 s), mis kinnitab galopi ühtlasemat perioodilisust.

5 Kokkuvõte

Käesoleva töö eesmärk oli kirjeldada hobuse ja ratsutaja interaktsiooni inertsiaalsensorite andmete põhjal ning hinnata, kuidas see interaktsioon muutub allüüri (samm/galopp) ja ratsutaja oskustaseme (algaja/oskaja) lõikes. Tulemused näitavad, et interaktsiooni korrelatsioon või antikorrelatsioon ja perioodilisus sõltub tugevalt allüürist – galopis avaldus signaalides selgem perioodiline struktuur ning nii mõõdetud signaalide aegridade graafikute kui korrelatsioonifunktsioonide põhjal olid hobuse–ratsutaja seosed ühtlasemad ja tõlgendatavamad. Sammu korral oli perioodilisus küll olemas, kuid korrelatsioonimustrid olid ebastabiilsemad ning hobuse–ratsutaja graafikud hajusamad, mis viitab kehvemale korreleeruvusele.

5.1 Allüüri mõju interaktsiooni struktuurile

Vertikaalkiirenduse a_z tulemused rõhutavad galopi ja sammude erinevust kõige selgemalt. Galopis olid ajasignaalid regulaarsed ning graafikud koherentsemad. See väljendus ka korrelatsioonides, kus autokorrelatsiooni funktsiooni maksimumide korduvus oli selgelt eristatav ja ristkorrelatsioonis esinesid tugevamad ning teravamad maksimumid. See tähendab, et galopis on võimalik hobuse ja ratsutaja ajastust (faasisuhet) hinnata robustsemalt, sest korduv perioodiline komponent domineerib ja juhuslik varieeruvus on väiksem.

Sammus seevastu olid a_z graafikud mõlema ratsutaja korral pigem laialivalguvad ning ristkorrelatsioonides esines mitu omavahel sarnase tugevusega tippu. Selline tulemus on biomehaaniliselt ootuspärane: samm on aeglasem ja vähem lukustatud rütmiga allüür ning väikesed tempo ja kehahoiaku muutused võivad tsükli kuju märgatavalt mõjutada. Selle tulemusena ei pruugi interaktsioon koonduda ühe kindla ajastuse ümber ning faasinihe sõltub rohkem sellest, milline lõik valitakse analüüsiks.

5.2 Oskustaseme mõju: kooskõla, amplituud ja stabiilsus

Algaja ja oskaja erinevused ilmnesevad kõige selgemalt galopis. Galopi a_z ajasignaalides oli algajal tendents suurematele ekstreemidele (eriti tugevamad sukeldumised), samas kui oskaja signaal oli sujuvama kujuga ja vähem äärmuslik. Sama järeldust toetas trajektoore kuju: oskaja hobuse–ratsutaja kiirendustrajektoor koondus kitsamaks ja korrapärasemaks kujundiks, algaja trajektoor kattis suuremat ala ja oli hajusam. Biomehaanilises tõlgenduses tähendab see, et oskaja liikumine on hobuse vertikaalse rütmiga stabiilsemalt seotud ning ratsutaja keha toimib efektiivsema summutina, mistõttu tsüklitevaheline varieeruvus väheneb. Algaja puhul võib suurem hajuvus viidata ebaühtlasele kohandumisele (mõnel tsükli jäänud maha, teisel kompenseerib teistmoodi), mis teeb hobuse ja ratsutaja signaalide suhte vähem püsivaks.

Sammu puhul oli algaja–oskaja vahe a_z põhjal nõrgemini eristuv. Kuigi oskaja trajektoor oli kohati koondunud, kattusid trajektoorid suurel määral ning korrelatsioonide põhjal ei eristunud üks selge dominantne ajastus. See tulemus ei tähenda, et oskustaseme erinevust sammus ei oleks, vaid pigem, et

valitud mõõdik (a_z) ja valitud analüüsimeetod (korrelatsioon lühikesel lõigul) on sammu korral tundlikumad varieeruvusele.

5.3 Kiirendus vs nurk: milline mõõdik töötab paremini?

Töö käigus selgus ka, et a_z ja θ_x ei käitu interaktsiooni kirjeldamisel ühtemoodi. Vertikaalkiirendus a_z on hea allüüri rütmi kandja - toob galopis hästi välja korrelatsiooni ning amplituudierinevusi. Samas sammus võib a_z olla liiga tundlik väikestele tempo- ja kehahoiaku muutustele ning seetõttu on interaktsioon graafikul ja ristkorrelatsioonis hajusam.

Nurk θ_x osutus kohati tundlikumaks ratsutaja stabiliseerimise ja kehahoiaku strateegia suhtes. Sammu θ_x graafikud ja korrelatsioonid eristasid algajat ja oskajat teistsugusel moel: erinevus ei avaldunud ainult amplituudis, vaid pigem seose hajuvuses ja koherentsuses. Galopis ilmnes θ_x korrelatsioonides huvitav nüanss: ristkorrelatsiooni tugevamad tipud ei paiknenud alati nulli juures, mis viitab sellele, et hobuse ja ratsutaja orientatsioonimuutus võib olla faasis nihkes (st parim kattuvus tekib ajanihega). See on loogiline, sest ratsutaja võib teadlikult või alateadlikult stabiliseerida keha nii, et orientatsioon ei ole hobuse kere suhtes täpselt samas faasis, vaid kompenseerib seda kindla nihkega. Selline faasisuhte tõlgendus on kooskõlas varasemate hobuse–ratsutaja interaktsiooni uuringute ideega, et interaktsioon ei avaldu ainult samal ajal liikumises, vaid ka stabiilses ajastus- ja faasisuhtes. [9]

5.4 Edasised töösuunad

Antud töö toimib hästi kui esimene samm hobuse–ratsutaja süsteemi interaktsiooni mõistmise suunas: meetoodika suudab eristada allüüri mõju ning näitab, et galopis on algaja ja oskaja erinevused paremini nähtavad, eriti a_z trajektooride ja ristkorrelatsiooni kaudu. Edasises töös annaks oluliselt tugevama aluse:

1. suurem valim (rohkem ratsutajaid, võimalusel rohkem hobuseid),
2. tempo ja lõikude valiku standardiseerimine (selged kriteeriumid stabiilsele lõigule),
3. faasinihke esitamine lisaks sekunditele ka suhtelise faasina (tsükli suhtes).

Kokkuvõtvalt näitavad tulemused, et inertsiaalsensoripõhine lähenemine võimaldab kirjeldada hobuse–ratsutaja interaktsiooni nii kvalitatiivselt kui ka kvantitatiivselt. Galopp annab seejuures puhtama signaali ja stabiilsema rütmi, mille pealt on interaktsiooninäitajad usaldusväärsemad. Samm on interaktsioon varieeruvam, mistõttu vajab kasutatud lähenemine rohkem standardiseerimist ja suuremat andmestikku.

6 Tõnuavaldused

Kõesoleva töö autor soovib tänada oma juhendajat Dmitri Kartofelevi juhendamise, konstruktivse kriitika ja toetuse eest. Lisaks soovib autor tänada katses osalenud algajana Kaisa Laksperet ning hobust Zahrazad, kes kannatlikult mitmed mõõtmisseansid läbis.

7 Annotatsioon

Käesoleva bakalaureusetöö eesmärk on katseliselt mõõta ja analüüsida hobuse ja ratsutaja interaktsiooni erinevates alüürides, kasutades inertsiaalsensoreid. Mõõtmised tehti maneežis sirgel lõigul sammus ja parema jalaga galopis sama hobusega, kuid kahe ratsutajaga (algaja ja oskaja), et vähendada hobusest tulenevat varieeruvust ning hinnata ratsutaja oskustaseme mõju interaktsioonile. Mõõteseadmetena kasutati Witmotion WT9011DCL-BT50 sensoreid proovivõtusagedusega 200 Hz - hobuse sensor kinnitati sadulavöö külge keskele alla ning ratsutaja sensor seljale. Analüüsis keskenduti vertikaalsele kiirendusele a_z (z-telg üles) ja x-suunalisele nurga komponendile θ_x .

Interaktsiooni kirjeldamiseks visualiseeriti hobuse ja ratsutaja signaalid ajas ning moodustati hobune–ratsutaja graafikud. Kvantitatiivseks hindamiseks arvutati signaalide autokorrelatsioonid (tsüklilisuse hindamiseks) ja ristkorrelatsioonid, mille maksimumi ajanihet tõlgendati faasinihkena. Tulemused näitasid, et galopis on signaalide perioodilisus ja korrelatsioonimustrid üldiselt selgemad kui sammus, mis võimaldab interaktsiooni ajastust usaldusväärsemalt hinnata. Oskaja ratsutaja korral olid graafikud mitmel juhul koherentsemad ja ristkorrelatsiooni maksimumid selgemini eristatavad, viidates stabiilsemale kohandumisele hobuse rütmiga. Töö piiranguks on väike valim (üks hobune ja kaks ratsutajat), mistõttu käsitletakse tulemusi esmaste tendentsidena. Edasises töös on vajalik suurem valim, tempo täpsem standardiseerimine ning metoodika laiendamine.

8 Abstract

The aim of this bachelor's thesis is to experimentally measure and analyze horse–rider interaction across different gaits using inertial measurement units. Measurements were conducted on a straight line in an indoor arena in walk and right-lead canter using the same horse with two riders (novice and experienced) in order to reduce horse-related variability and to assess the influence of rider skill on the interaction. Witmotion WT9011DCL-BT50 were used at a sampling rate of 200 Hz - the horse sensor was attached centrally to the girth (underside), and the rider sensor was mounted on the rider's back. The analysis focused on vertical acceleration a_z (z-axis upward) and the x-axis angle component θ_x .

Horse and rider signals were visualized in the time domain and by horse–rider trajectories to provide an intuitive representation of coupling. For quantitative assessment, autocorrelation functions were computed to characterize cyclicity, and cross-correlation functions were used to estimate interaction strength and timing; the time lag at the maximum cross-correlation was interpreted as a phase shift between horse and rider signals. The results indicate that canter exhibits clearer periodic structure and more distinct correlation patterns than walk, enabling more robust estimation of timing related interaction measures. In several cases, the experienced rider produced more coherent trajectories and more clearly identifiable cross-correlation maxima, suggesting a more stable adaptation to the horse's rhythm. The main limitation of this study is the small sample size (one horse and two riders) therefore the findings are presented as preliminary tendencies. Future work should include a larger dataset, stricter speed standardization, and an extended analysis framework.

9 Kasutatud kirjandus

- [1] Fédération Equestre Internationale (FEI), „FEI Dressage Judging Manual – Effective 1 January 2025“, [Võrgumaterjal].
<https://inside.fei.org/sites/default/files/FEI%20Dressage%20Judging%20Manual%20%20Effective%201%20January%202025.pdf>. [Kasutatud 30.12.2025].
- [2] V. Camomilla, E. Bergamini, S. Fantozzi ja G. Vannozzi, „Trends Supporting the In-Field Use of Wearable Inertial Sensors for Sport Performance Evaluation: A Systematic Review“, *Sensors*, kd. 18, nr. 3, art. 873, 2018. [Võrgumaterjal]. <https://doi.org/10.3390/s18030873>. [Kasutatud 30.12.2025].
- [3] C. M. Crecan, C. P. Peştean, „Inertial Sensor Technologies—Their Role in Equine Gait Analysis, a Review“, *Sensors*, kd. 23, nr. 14, art. 6301, 2023. [Võrgumaterjal].
<https://pmc.ncbi.nlm.nih.gov/articles/PMC10386433/>. [Kasutatud 30.12.2025].
- [4] C. Roepstorff, M.-T. Dittmann, S. Arpagaus, F. M. Serra Bragança, A. Hardeman, E. PerssonSjödin, L. Roepstorff, A. I. Gmel ja M. A. Weishaupt, „Reliable and clinically applicable gait event classification using upper body motion in walking and trotting horses“, *Journal of Biomechanics*, kd. 114, art. 110146, 2021, doi: 10.1016/j.jbiomech.2020.110146. [Võrgumaterjal]. <https://www.sciencedirect.com/science/article/pii/S0021929020305704>. [Kasutatud 31.12.2025].
- [5] Saitua Peñas, „Evaluation of the biomechanics of locomotion in the athletic horse by accelerometry“, doktoriõppe lõputöö, Universidad de Córdoba, 2023. [Võrgumaterjal].
<https://helvia.uco.es/xmlui/bitstream/handle/10396/24980/2023000002639.pdf>. [Kasutatud 30.12.2025].
- [6] L. Laffi, F. Bigand, C. Peham, G. Novembre, M. Gamba ja A. Ravignani, „Rhythmic categories in horse gait kinematics“, *Journal of Anatomy*, kd. 246, nr. 3, lk. 456–465, 2025, doi: 10.1111/joa.14200. [Võrgumaterjal].
<https://phaidra.vetmeduni.ac.at/api/object/o%3A3938/get>. [Kasutatud 30.12.2025].
- [7] H. M. Clayton, R. MacKechnie-Guire, S. J. Hobbs, „Riders' Effects on Horses-Biomechanical Principles with Examples from the Literature“, 2023, doi: 10.3390/ani13243854. [Võrgumaterjal]. <https://pmc.ncbi.nlm.nih.gov/articles/PMC10741103/>. [Kasutatud 02.01.2026].
- [8] H. M. Clayton, S. J. Hobbs, „The role of biomechanical analysis of horse and rider in equitation science“, 2017. [Võrgumaterjal].
<https://knowledge.lancashire.ac.uk/id/eprint/17182/1/17182%201-s2.0-S016815911730062Xmain.pdf>. [Kasutatud 06.01.2026].
- [9] Münz, F. Eckardt, K. Witte, „Horse-rider interaction in dressage riding“, *Human Movement Science*, kd. 33, lk. 227–237, 2014, doi: 10.1016/j.humov.2013.09.003.
- [10] J. Lagarde, C. Peham, T. Licka, J. A. S. Kelso, „Coordination dynamics of the horse-rider system“, 2005. [Võrgumaterjal]. <https://pmc.ncbi.nlm.nih.gov/articles/PMC1821095/#F3>. [Kasutatud 03.01.2026].
- [11] S. W. Smith, „Correlation“, teoses *The Scientist and Engineer's Guide to Digital Signal Processing*, ptk 13. [Võrgumaterjal]. <https://dspguide.com/ch13/2.htm>. [Kasutatud 02.01.2026].

- [12]NumPy Developers, „numpy.correlate“, *NumPy v2.3 Manual*, 2025. [Võrgumaterjal].
<https://numpy.org/doc/2.3/reference/generated/numpy.correlate.html>. [Kasutatud 05.01.2026].
- [13]SketchOk, „Horse and rider (silhouette)“, (pildifail). [Võrgumaterjal].
<https://sketchok.com/images/articles/07-animals/08-other/09/17m.jpg>. [Kasutatud 02.01.2026].
- [14]iStock (Getty Images), „English saddle / riders (vector)“, (pildifail). [Võrgumaterjal].
https://media.istockphoto.com/id/1183898049/vector/english-saddle-western-saddleriders.jpg?s=1024x1024&w=is&k=20&c=wJEwX2KkNLfO8DP2Q6pOTR_93kW1GN-SmewEQJbrY8M=. [Kasutatud 02.01.2026].
- [15]Tindie, „WT9011DCL-BT50 Bluetooth 5.0 9DOF angle sensor“, (tooteleht). [Võrgumaterjal].
<https://www.tindie.com/products/adz1122/wt9011dcl-bt50-bluetooth-50-9dof-anglesensor/>. [Kasutatud 02.01.2026].

10 Lisad

Lisa 1 Pythoni moodul fig0.py

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.integrate import cumulative_simpson

#_____parameters_____ juba_salvestatud = 1 #
kas_sorteeritud.npy file juba olemas, 1=True, 0=False.
ma_window = 1 #[samples] moving average algorithm. >= 1 save_plot
= 0 # if 1 = True saves plots to working directory else
sample_rate = 200 #[Hz] measurement samples per second. alyyr =
'galopp' tase = 'oskaja'

#_____selecting file to import below_____
if tase == 'oskaja': if alyyr == 'samm':
    raw_data_file = '20251108102715.txt' # oskaja samm 8 nov
elif alyyr == 'traav':
    raw_data_file = '20251108102849.txt' # oskaja traav 8 nov
elif alyyr == 'galopp': raw_data_file =
'20251108103013.txt' # oskaja galopp 8 nov else:
    raise Exception('alyyr valesti kirjutatud.')
elif tase == 'algaja': if alyyr == 'samm':
    raw_data_file = '20251108103433.txt' # algaja samm, 8 nov.
    elif alyyr == 'traav':
        raw_data_file = '20251108103547.txt' # algaja traav, 8 nov.
    elif alyyr == 'galopp': raw_data_file =
'20251108103634.txt' # algaja galopp, 8 nov. else:
        raise ValueError('alyyr valesti kirjutatud.')
else: raise ValueError('tase pole ei oskaja ega
algaja...')

#_____cut off frequency calculator_____ def
cf_calc(N, sr):
    '''Calculates cut off frequency in Hz. used to comment on
moving average low-pass filter.
    NB! 0.443 is the numeric solution of the equation for the
first -3 dB point of the Dirichlet kernel
    Parameters
    -----
    N : int | np.int
samples used in moving average algorithm.
    sr : int | np.int sampling rate in
original signal in Hz.
    Returns -----
cut-off frequency in Hz.
    '''
    if N <= 1:
        return None
    else:
        f_cf = 0.443 * sr / N
    return f_cf #_____signaalide
pikkuse uhtlustajad_____ def
to_same_len(s1, s2):
    ls1 = len(s1) ls2
= len(s2) if ls1 !=
ls2: if ls1 <
ls2: s2 =
```

```

s2[:ls1]          else:
s1 = s1[:ls2]      return
s1, s2

#_____kolme signaalide pikkuse uhtlustajad_____
def to_same_len3(s1, s2, s3):      l1, l2, l3 =
len(s1), len(s2), len(s3)      min_len = min(l1, l2,
l3)
    return s1[:min_len], s2[:min_len], s3[:min_len]

#_____autoCorr_____used in other modules
def autocor(x):      x = x - x.mean()
    result = np.correlate(x, x, mode='full')
    return result[result.size//2:]

#_____crossCorr_ver_2_____used in other modules def
lag_CC(x, y, t):
    '''Calcualtes cross correlation.      Also
    genetates negative time lag values,      in
    addition to crosscorr.      t (array) - half of lags
    time (0, max_time) '''      x = x - x.mean()      y =
y - y.mean()
    croscorr = np.correlate(x, y, mode='full')      rev_neg_t = -
t[:0:-1] #reversing, making negative, removing 0.
    t_lag = np.concatenate((rev_neg_t, t))
    return t_lag, croscorr

#_____crosscor_coefficient_____def
CCcoeff(x, y):
    '''calcualtes the Pearson correlation coefficient (single value)
    of crosscorrrelated signals. '''      coeff_mat = np.corrcoef(x, y)
    coeff = coeff_mat[0, 1] # voi [1, 0], corr x signaali ja y vahel.
    return coeff

#_____max corr value finder_____used in other modules def
max_CF(x, y):
    max_x = np.max(np.abs(x))      max_y =
np.max(np.abs(y))      value = max_x if max_x >
max_y else max_y      return value

#_____period_lines_____def
periodic_t(dispt, period, a, o):
    '''Arvutused mis vajalikud perioodi joonte kuvamiseks
    dispt (korteer) - sama mis plot_comp docstringis      period
    (korteer) - perioodi algus ja pikkus, vt. mooduli algus.
    a (np.array) - algaja ratsutaja signaal
    o (np.array) - oskaja hobune signaal
    RETURNS:
    p_n (np.array) - period line points (indeks [tk]).
    t_per (np.array) - time time-series in [s].      '''      if
dispt[1] == 0:
        if len(a) < len(o):
            t_per = a
    else:
        t_per = o
        p = np.arange(period[0], t_per[-1], period[1])
    else:
        if len(a) < len(o):
            t_per = a
    else:

```

```

        t_per = 0
        p = np.arange(period[0], t_per[-1], period[1])
    p_n = (p * sample_rate).astype('int')
    # Sometimes gives bigger
    if p_n[-1] > len(t_per):
        p_n = p_n[:-1]
    return p_n, t_per

#_____cropping timeline to sample n_____
def crop_dtn(displ_t, sample_rate):    '''used to
find cropped time indexes      displ_t (korteet) -
sama mis plot_comp docstringis      sample_rate (int)
- sample rate with Motion sensor
Returns:
    d_n0 (int) - time line start index.
    d_n (int) - time line end index.
'''
    if displ_t[1] == 0:            d_n0 =
int(round(displ_t[0] * sample_rate))    d_n =
-1    else:
        d_n0 = int(round(displ_t[0] * sample_rate))
    d_n = int(round(displ_t[1] * sample_rate))    return
    d_n0, d_n

#_____exporting asked x y z data_____ def
data_export(data, milline, ma_win, a_int_n=0, v0=0, v0comp='0', ang0=0,
ang0comp='0'):
    """Extracts the 1d arrays from sorted
and grouped 2D array.      muutuja milline
meaning:      1 --> kiirendus data
2 --> nurk sekundi kohta data
3 --> nurk, labi angel/s data integreerimise
4 --> nurk data, sensorist
    a_int_n: how many time to integrate acceleration signal does not effect
other signals      for milline = 1:
        v0 = 0 #[m/s] initial speed at t=0            v0comp = 'x' #direction of on
non-zero initial speed 'x','y','z' or '0'.
        for milline = 3:
            ang0 = 0            ang0comp = '0' # around what axis is the rotation 'x',
'y', 'z' or '0'.
    """

#_____moving average_____
def mov_aver(x, ma_win):
    """Moving average algorithm, ma_win            is window of sampel
size. Muudab pikkust aga data_export teab seda.
    """
    res = np.convolve(x, np.ones(ma_win), 'valid') / ma_win
    return res
    time = data[:,0].astype('float64')
    sc = data[:,1] # sensor code
    dt = 1/sample_rate #[s] time step, for theoretical time array      t
= np.arange(len(time)) * dt # before low pass filter      t_p =
np.arange(len(time) - ma_win + 1) * dt # after low pass filter      if
milline == 1:            g = 9.818235 #[m/s2] g in Tallinn            x = g *
data[:,2].astype('float64')            y = g * data[:,3].astype('float64')
z = g * data[:,4].astype('float64')
        # initial speed at t=0
    if v0comp != '0':
    if v0comp == 'x':

```

```

        v0tup = (v0*1000, 0, 0)
elif v0comp == 'y':
    v0tup = (0, v0*1000, 0)
elif v0comp == 'z':
    v0tup = (0, 0, v0*1000)
else:
    print('v0comp peab olema kas "0", "x", "y" voi "z"')
    # signal integration in time, velocity
if a_int_n == 1:
    x = cumulative_simpson(x, x=t, initial=v0tup[0]) * 1e-3
    y = cumulative_simpson(y, x=t, initial=v0tup[1]) * 1e-3
    z = cumulative_simpson(z, x=t, initial=v0tup[2]) * 1e-3
    # double integral in time, displacement
    if a_int_n == 2:
        x0 = cumulative_simpson(x, x=t, initial=v0tup[0])
        y0 = cumulative_simpson(y, x=t, initial=v0tup[1])
        z0 = cumulative_simpson(z, x=t, initial=v0tup[2])
        x = cumulative_simpson(x0, x=t, initial=0)
        y = cumulative_simpson(y0, x=t, initial=0)
        z = cumulative_simpson(z0, x=t, initial=0)
    elif milline == 2:
        x = data[:,5].astype('float64')
        y = data[:,6].astype('float64')
        z = data[:,7].astype('float64')
    elif milline == 3:
        x = data[:,5].astype('float64')
        y = data[:,6].astype('float64')
        z = data[:,7].astype('float64')
    if ang0comp != '0':
        if ang0comp == 'x':
            a0tup = (ang0, 0, 0)
        elif ang0comp == 'y':
            a0tup = (0, ang0, 0)
        elif ang0comp == 'z':
            a0tup = (0, 0, ang0)
        else:
            print('ang0comp peab olema kas "0", "x", "y" voi "z"')
            x = cumulative_simpson(x, x=t, initial=a0tup[0])
            y = cumulative_simpson(y, x=t, initial=a0tup[1])
            z = cumulative_simpson(z, x=t, initial=a0tup[2])
        else:
            x = cumulative_simpson(x, x=t, initial=0)
            y = cumulative_simpson(y, x=t, initial=0)
            z = cumulative_simpson(z, x=t, initial=0)
        elif milline == 4:
            x = data[:,8].astype('float64')
            y = data[:,9].astype('float64')
            z = data[:,10].astype('float64')
        else:
            print('milline peab olema kas 1, 2, 3, 4')
    return t_p, sc, mov_aver(x, ma_win), mov_aver(y, ma_win), mov_aver(z, ma_win)
# plotting asked data and saving figures
def plot_data(data, milline, plot_title, xaxl, yaxl, lx, ly, lz, ma_window,
int_n=0, v0=0, v0comp='0', ang0=0, ang0comp='0'):
    """Plots the sorted and grouped data,
for given data. three graph x, y, z
milline tahendus: 1 --> kiirendus
data
2 --> nurk sekundi kohta data
3 --> nurk data, integrated 4 --> nurk data, sensor
int_n: how many time to integrate acceleration signal does not effort other
signals for milline = 1:
v0 = 0 #[m/s] initial speed at t=0
v0comp = 'x' #direction of on non-zero initial speed 'x', 'y', 'z' or '0'
for milline = 3:
    ang0 = 0
    ang0comp = '0' # around what axis is the rotation 'x',
'y', 'z' or '0'.
    t, _, x, y, z = data_export(data, milline,

```

```

ma_window, int_n, v0,                                v0comp, ang0, ang0comp)
plt.figure()
    plt.plot(t, x, '-', label=lx)
plt.plot(t, y, '-', label=ly)
plt.plot(t, z, '-', label=lz)
plt.xlabel(xaxl)      plt.ylabel(yaxl)
plt.title(plot_title)
plt.legend(loc=0)
    if
save_plot:
    plt.savefig(f'Plots_fig0/{tase}/{alyyr}/{plot_title}_{tase}_{alyyr}.pdf',
bbox_inches='tight')      plt.show()

#_____code_separete sensors__sort__save sorted_____ if
__name__ == '__main__':
    faili_nimi = f'{raw_data_file.split(".")[0]}' #'Sorted_data_{raw_data_file}.npz'
if juba_salvestatud:      with open(f'data/Sorted_data_{faili_nimi}.npz', 'rb')
as f:      data_rat = np.load(f)      data_hob = np.load(f)      else:
    # data loading and format control
    t0,s1, f2,f3,f4, f5,f6,f7, f8,f9,f10 = np.loadtxt(f'data/{raw_data_file}',
dtype='S23,S24,d,d,d,d,d,d,d,d',      delimiter = '\t',
skiprows = 1,
    usecols = (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10),
unpack = True)
    # Use pandas to parse, then convert to numpy datetime64[ms]
t0 = t0.astype('str')
    dt64 = pd.to_datetime(t0, errors='coerce').to_numpy(dtype='datetime64[ms]')
    # time format dtype to float [ms] + data synchronisation      t = (dt64
- dt64[0]).astype('float64') # Eeldan et esimene aeg on esimene.
    # binary string to str
s1 = s1.astype('str')      #
stacking data into 2D array
    data = np.column_stack((t,s1, f2,f3,f4, f5,f6,f7, f8,f9,f10)) #type to str
# Split the array into two groups based on the second column=sensor name
sensore_names = np.unique(data[:, 1])
    data_groups = {key: data[data[:, 1] == key] for key in sensore_names}
    # data naming, dictionary keys
ratsutaja = sensore_names[0]      hobune
= sensore_names[1]
    # 2D array out of dictionary of data groups
data_rat = data_groups[ratsutaja]      data_hob
= data_groups[hobune]      # data sorting by
time column
    data_rat = data_rat[data_rat[:, 0].astype('float64').argsort()]
    data_hob = data_hob[data_hob[:, 0].astype('float64').argsort()]
    # saving sorted data into binary .npz file      with
open(f'data/Sorted_data_{faili_nimi}.npz', 'wb') as f:
np.save(f, data_rat)      np.save(f, data_hob)

```

Lisa 2 Pythoni moodul fig1.py

```
import numpy as np
import matplotlib.pyplot as plt
from fig0 import data_export, to_same_len, periodic_t, crop_dtn, cf_calc
#_____parameters_____
save_plot = 1 # [bool] 1 = True saves plots to working directory else no
sample_rate = 200 # [Hz] measurement samples per second. alyyr = 'samm'
disp_axis = 'z' disp_t = (0, 7) # [s] (algus lopp=0) how much timeseries
to display, 0 all
On_per_pulgad = 1 # kas paneme per pilgad vert.
period = (0.23, 0.58) # [s] (start, len_period). algaja period2
= (0.31, 0.59) # [s] (start, len_period). oskaja ma_win = 5
#[samples] moving average algorithm. >= 1

#_____ .numpy file data folder. 'Sorted_data_{faili_nimi}.numpy'_____ if
alyyr == 'samm':
    faili_nimi = '20251108102715' # oskaja samm hob ja rat
    faili_nimi2 = '20251108103433' # algaja samm elif alyyr ==
    'traav':
        faili_nimi = '20251108102849' # oskaja Traav hob ja rat
        faili_nimi2 = '20251108103547' # algaja traav elif alyyr ==
        'galopp':
            faili_nimi = '20251108103013' # oskaja galopp
            hob ja rat faili_nimi2 = '20251108103634' # algaja
        galopp else:
            raise ValueError('alyyr valesti
            kirjutatud!')

#_____ cheking stuff_____
assert 'data_export' in globals(), 'Malus pole funktsiooni data_export.' assert
'to_same_len' in globals(), 'Malus pole funktsiooni to_same_len.' assert
'periodic_t' in globals(), 'Malus pole funktsiooni periodic_t.' assert
'crop_dtn' in globals(), 'Malus pole funktsiooni crop_dtn.' assert 'cf_calc' in
globals(), 'Malus pole funktsiooni cf_calc.' if disp_t[0] > period[0]: raise
ValueError('Aegridade algused valesti valitud: disp_t[0] > period[0]') if
disp_t[1] != 0 and disp_t[1] < period[1]: raise ValueError('Aegridade lopud
valesti valitud: disp_t[1] < period[1]')

#_____ Loading saorted data_____
with open(f'data/Sorted_data_{faili_nimi}.numpy', 'rb') as f: # oskaja
    data_rat = np.load(f) data_hob = np.load(f)

with open(f'data/Sorted_data_{faili_nimi2}.numpy', 'rb') as f: # algaja
    data_rat2 = np.load(f) data_hob2 = np.load(f)

#_____ PLOT def_____
def plot_comp(disp_t, dRat, dHob, dRat2, dHob2, milline, disp_axis, title,
xl, yl, lab, ma_win=1, int_n=0, v0=0, v0comp='0', ang0=0,
ang0comp='0'):
    """Using the sorted and grouped data,
    for given data. finds for X_x, X_y, X_z
    calculates:
        np.convolve(in1, in2) np.correlate(in1, in2) and np.correlate(in1, in1)
    disp_t - time in seconds how much of timeseries to display
    disp_axis = angle in horse framework, umber mis telje nurka kuvada, 'x', 'y',
```

```

'z'      muutuja milline
meaning:      1 -->
kiirendus data
2 --> nurk sekundi kohta data
3 --> nurk, labi angel/s data integreerimise
4 --> nurk data, sensorist      a_int_n: how many times to integrate acceleration
      signal.
      signals      for
milline = 1:
      v0 = 0 #[m/s] initial speed at t=0      v0comp = 'x' #direction of
non-zero initial speed 'x', 'y', 'z' or '0'.      for milline = 3:
      ang0 = 0      ang0comp = '0' # around what axis is non-zero angl 'x',
'y', 'z' or '0'.

      Pea meeles: probleemi geomeetria. hobuse teljed globaalsed.
ratsutaja      hobene      z      <=      y      -y      <=      z
x-90      <=      x
      teostatud all sonaraamatutes
      ""
      lw = 0.9 # Line width of main graphs
      d_n0, d_n = crop_dtn(dispt, sample_rate) # cropping timeline

      # algaja ja hobune
      tup_a = data_export(dRat2, milline, ma_win, int_n, v0, v0comp, ang0[0],
ang0comp[0])
      tup_ha = data_export(dHob2, milline, ma_win, int_n, v0, v0comp, ang0[1],
ang0comp[1])
      a = {'t':tup_a[0], 'x':tup_a[2]-90, 'z':tup_a[3], 'y':-tup_a[4]} #algaja
ha = {'t':tup_ha[0], 'x':tup_ha[2], 'y':tup_ha[3], 'z':tup_ha[4]} #hobune
      # oskaja ja hobune problem geometry horse framework primary      tup_o
= data_export(dRat, milline, ma_win, int_n, v0, v0comp, ang0[0],
ang0comp[0])
      tup_ho = data_export(dHob, milline, ma_win, int_n, v0, v0comp, ang0[1],
ang0comp[1])
      o = {'t':tup_o[0], 'x':tup_o[2]-90, 'z':tup_o[3], 'y':-tup_o[4]} #oskaja
ho = {'t':tup_ho[0], 'x':tup_ho[2], 'y':tup_ho[3], 'z':tup_ho[4]} #hobune
      plt.figure(figsize=(7.5, 9))
#algaja: nurk(aeg)      plt.subplot(3,
2, 1)
      plt.plot(a['t'][d_n0:d_n], a[disp_axis][d_n0:d_n], 'b-', lw=lw, label=lab[2])
plt.plot(ha['t'][d_n0:d_n], ha[disp_axis][d_n0:d_n], 'k-', lw=lw, label=lab[3])      if
On_per_pulgad:
      p_n, t_per = periodic_t(dispt, period, a['t'][d_n0:d_n], ha['t'][d_n0:d_n])
for i in p_n:      plt.axvline(x=t_per[i], lw=0.5, c='k', ls='--')
plt.xlabel(xl[0])      plt.ylabel(yl[0])      #plt.title(title[0])
plt.legend(loc=0)

      #algaja: nurk(nurk)      plt.subplot(3, 2, 2) a_s, ha_s =
to_same_len(a[disp_axis], ha[disp_axis]) # makes equal lenght
plt.plot(a_s[d_n0:d_n], ha_s[d_n0:d_n], 'g-', lw=lw, label=lab[2])
plt.xlabel(xl[1]) plt.ylabel(yl[1])      #plt.title(title[1])
plt.legend(loc=0)

      #oskaja: nurk(aeg)
plt.subplot(3, 2, 3)
      plt.plot(o['t'][d_n0:d_n], o[disp_axis][d_n0:d_n], 'b-', lw=lw, label=lab[0])
plt.plot(ho['t'][d_n0:d_n], ho[disp_axis][d_n0:d_n], 'k-', lw=lw, label=lab[1])
if On_per_pulgad:      p_n, t_per = periodic_t(dispt, period2,

```

```

o['t'][d_n0:d_n], ho['t'][d_n0:d_n])          for i in p_n:
plt.axvline(x=t_per[i], lw=0.5, c='k', ls='-')    plt.xlabel(xl[0])
plt.ylabel(yl[0])      #plt.title(title[0])      plt.legend(loc=0)

#oskaja: nurk(nurk)
plt.subplot(3, 2, 4)
    o_s, ho_s = to_same_len(o[disp_axis], ho[disp_axis]) # makes equal lenght
plt.plot(o_s[d_n0:d_n], ho_s[d_n0:d_n], 'r-', lw=lw, label=lab[0])
plt.xlabel(xl[1])      plt.ylabel(yl[1])      #plt.title(title[1])
plt.legend(loc=0)

#oskaja ja algaja: nurk(nurk)
plt.subplot(3, 2, (5, 6))
    plt.plot(a_s[d_n0:d_n], ha_s[d_n0:d_n], 'g-', lw=lw, label=lab[2])
plt.plot(o_s[d_n0:d_n], ho_s[d_n0:d_n], 'r-', lw=lw, label=lab[0])
plt.xlabel(xl[1])      plt.ylabel(yl[1])      #plt.title(title[1])
plt.legend(loc=0)

plt.tight_layout()
if save_plot:
    plt.savefig(f'Plots_fig1/{alyyr}/{title[1]}_axis_{disp_axis}_{alyyr}.pdf',
bbox_inches='tight')    plt.show()

#_____plotting plot_____ title
= ('Nurk vs aeg', 'Nurk vs nurk')
xl = (r'Aeg $t$ [s]', r'Ratsutaja nurk [$^{\circ}$]) # axis labels x
yl = (r'Nurk [$^{\circ}$]', r'Hobuse nurk [$^{\circ}$]) # axis labels y
lab = ('Oskaja', 'Hobune', 'Algaja', 'Hobune') # legend labels #
algtingimused nulllist erinevale teljele (ratsutaja, hobune)
ang0 = (90, 0) #ratsutaja nulllist erinev algnurk ja kobuse nulllist erinev = 0
ang0comp = ('x', '0') # hobuse 0 tahendab ara muuda midagi
plot_comp(disp_t, data_rat, data_hob, data_rat2, data_hob2, 3, disp_axis, title,
xl, yl, lab, ma_win, int_n=0, v0=0, v0comp='0', ang0=ang0,
ang0comp=ang0comp)
    if ma_win >=
1:
    print("--> Cut-off freq.", cf_calc(ma_win, sample_rate) ,"Hz (-3 dB drop)") if
On_per_pulgad:
    print(f"--> Algaja perioodile {period[1]} s vastav sag. {1/period[1]} Hz")
    print(f"--> Oskaja perioodile {period2[1]} s vastav sag. {1/period2[1]} Hz")

```

Lisa 3 Pythoni moodul fig2.py

```

import numpy as np import
matplotlib.pyplot as plt
from fig0 import data_export, to_same_len3, autocor, max_CF from
fig0 import periodic_t, crop_dtn, lag_CC, cf_calc, CCcoeff

#_____parameters_____
save_plot = 1 #[bool] 1 = True saves plots to working directory else no
sample_rate = 200 #[Hz] measurement samples per second. alyyr = 'samm'
disp_axis = 'x' disp_t = (0, 7) #[s] (algus lopp=0) how much timeseries
to display, 0 all
On_per_pulgad = 1 # kas paneme per pilgad vert.

```

```

period = (0.0, 0.64) #[s] (start, len_period). algaja period2
= (0.0, 0.572) #[s] (start, len_period). oskaja
period3 = (0.62, period[1]) #[s] CC graafiku nihe negatiivses suunas # algaja
period4 = (0.52, period2[1]) #[s] CC graafiku nihe negatiivses suunas # oskaja

ma_win = 5 #[samples] moving average algorithm. >= 1

#_____ .npz file data folder. 'Sorted_data_{faili_nimi}.npz'_____
if alyyr == 'samm': faili_nimi = '20251108102715' # oskaja samm hob ja
rat faili_nimi2 = '20251108103433' # algaja samm elif alyyr ==
'traav':
    faili_nimi = '20251108102849' # oskaja Traav hob ja rat
faili_nimi2 = '20251108103547' # algaja traav elif alyyr ==
'galopp': faili_nimi = '20251108103013' # oskaja galopp
hob ja rat faili_nimi2 = '20251108103634' # algaja
galopp else:
    raise ValueError('alyyr valesti kirjutatud!')

#_____ cheking stuff_____
assert 'data_export' in globals(), 'Malus pole funktsiooni data_export.'
assert 'to_same_len3' in globals(), 'Malus pole funktsiooni to_same_len3.'
assert 'autocor' in globals(), 'Malus pole funktsiooni autocor.' assert
'max_CF' in globals(), 'Malus pole funktsiooni max_CF.' assert
'periodic_t' in globals(), 'Malus pole funktsiooni periodic_t.' assert
'crop_dtn' in globals(), 'Malus pole funktsiooni crop_dtn.' assert
'lag_CC' in globals(), 'Malus pole funktsiooni lag_CC.' assert 'cf_calc'
in globals(), 'Malus pole funktsiooni cf_calc.' assert 'CCcoeff' in
globals(), 'Malus pole funktsiooni CCcoeff.' if disp_t[0] > period[0]:
    raise ValueError('Aegridade algused valesti valitud: disp_t[0] > period[0]')
if disp_t[1] != 0 and disp_t[1] < period[1]:
    raise ValueError('Aegridade lopud valesti valitud: disp_t[1] < period[1]')
#_____ Loading saorted data_____
with open(f'data/Sorted_data_{faili_nimi}.npz', 'rb') as f: # oskaja
data_rat = np.load(f) data_hob = np.load(f)

with open(f'data/Sorted_data_{faili_nimi2}.npz', 'rb') as f: # algaja
    data_rat2 = np.load(f) data_hob2 = np.load(f)

#_____ PLOT def_____
def plot_comp(disp_t, dRat, dHob, dRat2, dHob2, milline, disp_axis, title, xl,
yl, lab, ma_win=1, int_n=0, v0=0, v0comp='0', ang0=0, ang0comp='0'):
    """Using the sorted and grouped data, for given data. finds for X_x,
X_y, X_z calculates: np.convolve(in1, in2) np.correlate(in1, in2)
and np.correlate(in1, in1)

    disp_t - time in seconds how much of timeseries to display
    disp_axis = angle in horse framework, umber mis telje nurka kuvada, 'x', 'y'
                'z'
    muutuja milline meaning:
    1 --> kiirendus data
    2 --> nurk sekundi kohta data
    3 --> nurk, labi angel/s data integreerimise
    4 --> nurk data, sensorist a_int_n: how many times to integrate acceleration
        signal.
        signals for
    milline = 1:

```

```

        v0 = 0 #[m/s] initial speed at t=0          v0comp = 'x' #direction of
non-zero initial speed 'x', 'y', 'z' or '0'.      for milline = 3:
        ang0 = 0          ang0comp = '0' # around what axis is non-zero angl 'x',
'y', 'z' or '0'.

        Pea meeles: probleemi geomeetria. hobuse teljed globaalsed.
        ratsutaja      hobune
z      <=      y      -y
<=      z      x-90      <=
x
        teostatud all sonaraamatutes
        ""
        lw = 0.9 # Line width of main graphs
        d_n0, d_n = crop_dtn(dispt, sample_rate) # cropping timeline

        # algaja ja hobune
        tup_a = data_export(dRat2, milline, ma_win, int_n, v0, v0comp, ang0[0],
ang0comp[0])
        tup_ha = data_export(dHob2, milline, ma_win, int_n, v0, v0comp, ang0[1],
ang0comp[1])
        a = {'t':tup_a[0], 'x':tup_a[2]-90, 'z':tup_a[3], 'y':-tup_a[4]} #algaja
ha = {'t':tup_ha[0], 'x':tup_ha[2], 'y':tup_ha[3], 'z':tup_ha[4]} #hobune
        # oskaja ja hobune problem geometry horse framework primary      tup_o
= data_export(dRat, milline, ma_win, int_n, v0, v0comp, ang0[0],
ang0comp[0])
        tup_ho = data_export(dHob, milline, ma_win, int_n, v0, v0comp, ang0[1],
ang0comp[1])
        o = {'t':tup_o[0], 'x':tup_o[2]-90, 'z':tup_o[3], 'y':-tup_o[4]} #oskaja
ho = {'t':tup_ho[0], 'x':tup_ho[2], 'y':tup_ho[3], 'z':tup_ho[4]} #hobune

        plt.figure(figsize=(7.5, 7.2))

        #algaja ja oskaja: autocorr()
        plt.subplot(2, 2, 1)
        plt.axhline(0, lw=0.5, c='k', ls='-')      a_sig = a[disp_axis][d_n0:d_n]
        # algaja      o_sig = o[disp_axis][d_n0:d_n] # oskaja      mCF =
        max_CF(autocor(a_sig), autocor(o_sig)) plt.plot(a['t'][d_n0:d_n],
autocor(a_sig)/mCF, 'b-', lw=lw, label=lab[2]) plt.plot(o['t'][d_n0:d_n],
autocor(o_sig)/mCF, 'k-', lw=lw, label=lab[0])
        if On_per_pulgad:      p_n, t_per = periodic_t(dispt, period,
a['t'][d_n0:d_n], o['t'][d_n0:d_n])      for i in p_n:
            plt.axvline(x=t_per[i], lw=0.5, c='b', ls='-')
            p_n, t_per = periodic_t(dispt, period2, a['t'][d_n0:d_n], o['t'][d_n0:d_n])
        for i in p_n:      plt.axvline(x=t_per[i], lw=0.5, c='k', ls='-')
        plt.xlabel(xl)      plt.ylabel(yl[0])      #plt.title(title[0])      plt.legend(loc=0)

        #algaja hob ja oskaja hob: autocorr()
        plt.subplot(2, 2, 2)
        plt.axhline(0, lw=0.5, c='k', ls='-')
        ha_sig = ha[disp_axis][d_n0:d_n] #algaja
ho_sig = ho[disp_axis][d_n0:d_n] #oskaja      mCF
= max_CF(autocor(ha_sig), autocor(ho_sig))
        plt.plot(ha['t'][d_n0:d_n], autocor(ha_sig)/mCF, 'b-', lw=lw, label=lab[3])
        plt.plot(ho['t'][d_n0:d_n], autocor(ho_sig)/mCF, 'k-', lw=lw, label=lab[1])
        if On_per_pulgad:      p_n, t_per = periodic_t(dispt, period,
ha['t'][d_n0:d_n], ho['t'][d_n0:d_n])      for i in p_n:
            plt.axvline(x=t_per[i], lw=0.5, c='b', ls='-')      p_n, t_per =
periodic_t(dispt, period2, ha['t'][d_n0:d_n], ho['t'][d_n0:d_n])      for i
in p_n:      plt.axvline(x=t_per[i], lw=0.5, c='k', ls='-')

```

```

plt.xlabel(xl)      plt.ylabel(yl[0])      #plt.title(title[1])
plt.legend(loc=0)

#algaja ja oskaja: crosscorr(x, x_hobune)
plt.subplot(2, 2, (3, 4))      plt.axhline(0,
lw=0.5, c='k', ls='-')
a_t, a_sig, ha_sig = to_same_len3(a['t'][d_n0:d_n], a_sig, ha_sig)
o_t, o_sig, ho_sig = to_same_len3(o['t'][d_n0:d_n], o_sig, ho_sig)      mCF
= max_CF(lag_CC(a_sig, ha_sig, a_t), lag_CC(o_sig, ho_sig, o_t))      a_t2
= lag_CC(a_sig, ha_sig, a_t)[0] # time lag has negative values      a_CC =
lag_CC(a_sig, ha_sig, a_t)[1] # crosscorr
CC_coeff_algaja = CCcoeff(a_sig, ha_sig)

plt.plot(a_t2, a_CC/mCF, 'b-', lw=lw, label=lab[4])
o_t2 = lag_CC(o_sig, ho_sig, o_t)[0] # time lag has negative values
o_CC = lag_CC(o_sig, ho_sig, o_t)[1] # crosscorr
CC_coeff_oskaja = CCcoeff(o_sig, ho_sig)
plt.plot(o_t2, o_CC/mCF, 'k-', lw=lw,
label=lab[5])      if On_per_pulgad:
p_n, t_per = periodic_t(dispt, period3, a_t, o_t)
for i in p_n:
plt.axvline(x=-t_per[i], lw=0.5, c='b', ls='-')
plt.axvline(x=t_per[i] - 2*period3[0], lw=0.5, c='b', ls='-')
p_n, t_per = periodic_t(dispt, period4, a_t, o_t)      for i in p_n:
plt.axvline(x=-t_per[i], lw=0.5, c='k', ls='-')
plt.axvline(x=t_per[i] - 2*period4[0], lw=0.5, c='k', ls='-')
plt.xlabel(xl) plt.ylabel(yl[1])
plt.title(title[2])
plt.legend(loc=0)

plt.tight_layout()
if save_plot:
plt.savefig(f'Plots_fig2/{alyyr}/CF_nurk_axis_{dispt}_{alyyr}.pdf',
bbox_inches='tight')      plt.show()
return CC_coeff_algaja, CC_coeff_oskaja

# _____ plotting plot _____ title
= ('Autokorrelatsioon ratsanik', 'Autokorrelatsioon hobune',
'Ristkorrelatsioon ratsanik ja hobune') xl
= r'Aeg $t$ [s]' # axis labels x
yl = (r'Autokorrelatsioon [1]', r'Ristkorrelatsioon [1]')
lab = ('Oskaja', 'Oskaja hobune', 'Algaja', 'Algaja hobune', r'Algaja$\star$hobune',
r'Oskaja$\star$hobune') # legend labels

# algttingimused nulllist erinevale teljele (ratsutaja, hobune)
ang0 = (90, 0) #ratsutaja nulllist erinev algnurk ja kobuse nulllist erinev = 0
ang0comp = ('x', '0') # hobuse 0 tahendab ara muuda midagi
CCc_a, CCc_o = plot_comp(dispt, data_rat, data_hob, data_rat2, data_hob2, 3,
dispt, title, xl, yl, lab, ma_win, int_n=0, v0=0,
v0comp='0', ang0=ang0, ang0comp=ang0comp)
if ma_win >=
1:
print("--> Cut-off freq.", cf_calc(ma_win, sample_rate), "Hz (-3 dB drop)") if
On_per_pulgad:
print(f"--> Algaja perioodile {period[1]} s vastav sag. {1/period[1]} Hz")
print(f"--> Oskaja perioodile {period2[1]} s vastav sag. {1/period2[1]} Hz")
print('crosscorrelation coefficet algaja:',
CCc_a) print('crosscorrelation coefficet oskaja:',
CCc_o)

```

Lisa 4 Pythoni moodul `fig3.py` ja `fig4.py`

Moodulid `fig3.py` ja `fig4.py` on sisult sarnased moodulitele `fig1.py` ja `fig2.py`. Erinevus seisned ainult selles milliseid aegridu kuvatakse. Sellest lähtuvalt ei lisata nende sisu sellele dokumendile.

Lisa 5 Lihtlitsents

Lisa
rektori 07.04.2020 käskkirjale nr 1-8/17

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks¹

Mina, Kirke Ilves,

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose HOBUSE JA RATSANIKU INTERAKTSIOONI DÜNAAMIKA ANALÜÜS, mille juhendaja on Dmitri Kartofelev,
 - 1.1 reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
 - 1.2 üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

06.01.2026

¹ Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.