

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS

Artūras Vyčas

**TRANSPORTO PRIEMONIŲ GREIČIO NUSTATYMO
ALGORITMŲ, PANAUDOJANT MAGNETINIO LAUKO
JUTIKLIŲ DUOMENIS, SUKŪRIMAS IR TYRIMAS**

Baigiamasis magistro projektas

Vadovas:
Prof. D. Navikas

KAUNAS, 2018

KAUNO TECHNOLOGIJOS UNIVERSITETAS
ELEKTROS IR ELEKTRONIKOS FAKULTETAS
ELEKTRONIKOS INŽINERIJOS KATEDRA

**TRANSPORTO PRIEMONIŲ GREIČIO NUSTATYMO
ALGORITMŲ, PANAUDOJANT MAGNETINIO LAUKO
JUTIKLIŲ DUOMENIS, SUKŪRIMAS IR TYRIMAS**

Baigiamasis magistro projektas

Elektronikos inžinerija (kodas 621H61002)

Vadovas

(parašas) Prof. Dangirutis Navikas
(data)

Recenzentas

(parašas) Prof. Darius Andriukaitis
(data)

Projektą atliko

(parašas) Artūras Vyčas
(data)

KAUNAS, 2018



KAUNO TECHNOLOGIJOS UNIVERSITETAS

Elektros ir elektronikos

(Fakultetas)

Artūras Vyčas

(Studento vardas, pavardė)

Elektronikos inžinerija 621H61002

(Studijų programos pavadinimas, kodas)

Baigiamojo projekto

„Transporto priemonių greičio nustatymo algoritmų, panaudojant magnetinio lauko jutiklių duomenis, sukūrimas ir tyrimas“

AKADEMINIO SAŽINGUMO DEKLARACIJA

20 18 m. gegužės 28 d.
Kaunas

Patvirtinu, kad mano **Artūro Vyčo** baigiamasis projektas tema „Transporto priemonių greičio nustatymo algoritmų, panaudojant magnetinio lauko jutiklių duomenis, sukūrimas ir tyrimas“ yra parašytas visiškai savarankiškai, o visi pateikti duomenys ar tyrimų rezultatai yra teisingi ir gauti sąžiningai. Šiame darbe nei viena dalis nėra plagijuota nuo jokių spausdintinių ar internetinių šaltinių, visos kitų šaltinių tiesioginės ir netiesioginės citatos nurodytos literatūros nuorodose. Įstatymų nenumatytų piniginių sumų už šį darbą niekam nesu mokėjęs.

Aš suprantu, kad išaiškėjus nesąžiningumo faktui, man bus taikomos nuobaudos, remiantis Kauno technologijos universitete galiojančia tvarka.

(vardą ir pavardę įrašyti ranka)

(parašas)

Vyčas, Artūras. Transporto priemonių greičio nustatymo algoritmų, panaudojant magnetinio lauko jutiklių duomenis, sukūrimas ir tyrimas. Elektronikos inžinerijos *magistro* baigiamasis projektas / vadovas prof. Dangirutis Navikas; Kauno technologijos universitetas, Elektros ir elektronikos fakultetas, Elektronikos inžinerijos katedra.

Mokslo kryptis ir sritis: Elektros ir elektronikos inžinerija, Technologiniai mokslai

Reikšminiai žodžiai: TP greičio nustatymas, , AMR sensorius, transporto priemonė, vėlinimo tarp signalų nustatymas.

Kaunas, 2018. 77 p.

SANTRAUKA

Pagrindinis šio darbo tikslas – išbandyti įvairius vėlinimo tarp dviejų signalų radimo būdus, įvertinti jų privalumus bei trūkumus ir pasiūlyti naują algoritmą, kurio dėka būtų galima apskaičiuoti pravažiuojančios transporto priemonės greitį su bendros paskirties mikrovaldikliu. Skaičiavimai turėtų būti atlikti kaip įmanoma greičiau, o transporto priemonės greitis būtų apskaičiuojamas neviršijant užsiduotos paklaidos. Duomenų surinkimui naudojama sistema sudaryta iš dviejų AMR tipo magnetinio lauko jutiklių, mikrovaldiklio, kuris nuskaito duomenis iš sensorių, juos apdoroja ir siunčia duomenis į pagrindinį kompiuterį.

Pirmojo šio darbo dalyje analizuojama literatūra ir pateikiamas įvairių būdų, kuriuos naudojant galima apšvalga aptikti transporto priemonės, apžvalga. Taip pat apžvelgiami transporto priemonių klasifikavimo būdai naudojant tuos pačius duomenis.

Antroje darbo dalyje apžvelgiami įvairūs eksperimentiškai išbandyti vėlinimo tarp dviejų signalų radimo būdai, aptariama šių būdų pritaikymo galimybė signalų moduliams bei Z komponentėms, aptariama slenkančio vidurkio įtaka rezultatams, tiriamos skaičiavimo rezultatų patikslinimo galimybės.

Trečioje darbo dalyje apžvelgiami eksperimentiniai rezultatai kurie atlikti „Matlab“ programiniu paketu bei mikrovaldiklyje. Panaudojant gautus rezultatus sukurtas algoritmas, kuris gali būti taikomas siekiant gauti užsiduotą skaičiavimo tikslumą esant mažiausiai skaičiavimo trukmei.

Vyčas, Artūras. Research and Creation of Vehicle Speed Detection Algorithms Using Magnetic Field Sensors. *Master's thesis in Electronics Engineering / supervisor assoc. prof. Dangirutis Navikas. Kaunas University of Technology, Faculty of Electrical and Electronics Engineering, Department of Electronics Engineering*

Research area and field: Electrical and Electronics Engineering, Technological Sciences

Key words: Vehicle speed detection, AMR sensor, Vehicle, Find lag between two signals, Kaunas, 2018. 77 p.

SUMMARY

The main objective of this work – test various methods to find delay between two signals, evaluate their advantages and disadvantages and propose a new algorithm which would allow to calculate the speed of passing vehicle with any general purpose microcontroller. Calculations should be done as quickly as possible, speed measurement error should be in bounds with allowed error. The system used for collecting data is made of two AMR type magnetic field sensors, microcontroller, which reads data from the sensors and does the processing and sends the data to the main computer.

In first part of this work literature analysis is done and various methods to detect vehicle are discussed. Also various methods for classifying vehicles are discussed. Classification is done by using same data from AMR type magnetic field sensors.

In second part of this work various methods are tested experimentally which allow to find delay between two signals. There is also a feasibility discussion of applying discussed methods to signal modules and Z components. Finally, the influence of moving average to speed measurement error and feasibility of refining down measurement results are discussed.

In final part of this work test results, which are done with „Matlab“ and real MCU are discussed. Based on results, new algorithm is proposed which allows to achieve reasonable measurement errors with lowest possible calculation time.

TURINYS

ĮVADAS	8
1. LITERATŪROS ANALIZĖ.....	10
1.1 TP aptikimas.....	11
1.1.1 TP aptikimas AMR tipo magnetinio lauko jutikliais	15
1.2 TP klasifikavimas	20
1.3 TP greičio nustatymas.....	25
2. VĒLINIMO TARP DVIEJŲ SIGNALŲ NUSTATYMO BŪDAI	30
2.1 Kros-Koreliacija.....	31
2.2 Masės centro radimas	33
2.3 Figūros, apribotos slenkstinėmis vertėmis, centro skaičiavimas.....	34
2.4 Slenkstinės reikšmės viršijimo metodas	35
2.5 Atstumo tarp maksimumų skaičiavimas.....	36
2.6 Mažiausio kvadratų skirtumo metodas ir mažiausio skirtumo skaičiavimas.....	37
2.7 Vienodų plotų metodas	39
2.8 Slenkstinės vertės viršijimo radimas, kai slenkstis lygus pirmojo signalo vidurkiui	40
2.9 Figūros, apribotos slenkstinėmis vertėmis iš abiejų pusių, vidurkio skaičiavimas	41
2.10 Figūros, apribotos slenkstinėmis vertėmis, centrų skirtumo skaičiavimas	42
2.11 Suderinto filtro dažnių srityje metodas.....	43
2.12 Dalinės kros-koreliacijos skaičiavimas.....	44
2.13 Vėlinimo radimo būdų tarp dviejų signalų apibendrinimas	46
2.14 Vidurkinimo įtaka skaičiavimų rezultatams	47
2.15 Rezultatų patikslinimo būdai	50
2.16 Vėlinimo tarp dviejų signalų skaičiavimo būdų pritaikymas Z ašiai	51
3. EKSPERIMENTINIAI REZULTATAI.....	54
3.1 Vėlinimo tarp signalų skaičiavimas mikrovaldiklyje.....	58
IŠVADOS IR PASIŪLYMAI	65
INFORMACIJOS ŠALTINIŲ SĄRAŠAS	66
PRIEDAI.....	68
Priedas 1. MATLAB programa	68

SANTRUMPŲ IR ŽENKLŲ AIŠKINIMO ŽODYNAS

AMR	Anizotropinis magnetorezistyvnis (angl. Anisotropic Magnetoresistance)
CC	Kros-koreliacija (angl. Cross Correlation)
FFT	Greitoji Furjė transformacija (angl. Fast Fourier Transform)
MCU	Mikrovaldiklis (angl. Microcontroller Unit)
MEMS	Mikro elektro mechaninė sistema (angl. Microelectromechanical System)
SMA	Slenkantis vidurkis (angl. Simple Moving Average)
STDEV	Standartinė deviacija (angl. Standard Deviation)
TP	Transporto Priemonė

IVADAS

Šiuolaikinės ir modernios transporto priemonių srautų valdymo sistemos surenka įvairiausius duomenis: pravažiavusių transporto priemonių skaičių, jų greitį, transporto priemonės tipą. Dar daugiau, norint transporto priemones suklasifikuoti, turėtume žinoti transporto priemonės ilgį. Visa tai įmanoma atlikti tik tuomet, kai galime tiksliai išmatuoti transporto priemonės greitį.

Transporto priemonių greičiui nustatyti yra daugybė sistemų, kurių veikimo principai skirtingi. Pavyzdžiui, Doplerio efektu paremtos sistemos, arba LIDAR radarai. Taip pat transporto priemonių greičiui matuoti gali būti naudojamos sistemos, kuriose vaizdo kameros fiksuoja pravažiuojančias transporto priemones (angl. SDCS – *Speed Detection Camera System*) [1]. Tuomet daromas nufilmuotos medžiagos vaizdų apdorojimas ir pagal tai nustatomas transporto priemonės greitis.

Dar vienas sistemų tipas, kuris gali būti naudojamas transporto priemonių greičio nustatymui yra taip vadinami automatiniai transporto srauto klasifikatoriai (angl. ATCs – *Automated Traffic Classifiers*), kuriuos sudaro indukcinės kilpos, kurios montuojamos po važiuojamąja kelio dalimi [2]. Tokios sistemos būna brangios nes reikalauja papildomų adapterių duomenų apdorojimui, be to kiekvieną kartą norint sumontuoti sistemą reikia gręžti po asfaltu, kas užima daug laiko ir yra brangi procedūra, be to reikia gauti tam tikrus leidimus iš atitinkamų institucijų.

Paskutiniu metu daugybė tyrimų šioje srityje atliekama naudojant magnetinio lauko jutiklius. Matavimo principas, yra gana paprastas – po važiuojamąjį kelio dalimi, arba šalia jos, sumontuojami du magnetinio lauko jutikliai tam tikru atstumu vienas nuo kito. Atstumas tarp jutiklių iš anksto būna žinomas. Tyrimai šioje kryptyje rodo, kad naudojant magnetinio lauko jutiklius galima apskaičiuoti ne tik pravažiuojančios transporto priemonės greitį, bet ir nustatyti jos ilgį, ar identifikuoti kokio tipo tai transporto priemonė pagal unikalų parašą, kurį automobilis palieka pravažiavęs tarp dviejų magnetinio lauko jutiklių.

Pagrindinė problema ta, kad norint apskaičiuoti transporto priemonės greitį, reikia surasti vėlinimą tarp dviejų signalų, kuriuos sukuria tarp dviejų magnetinio lauko jutiklių pravažiavusi transporto priemonė. Ta galima padaryti įvairiais metodais, tačiau vieni metodai yra lėtesni už kitus, kitiems reikia daugiau mikrovaldiklio resursų, kiti yra ne tokie tikslūs ir jų nepakanka norint teisingai įvertinti transporto priemonės greitį dėl gaunamų per didelių paklaidų.

Pagrindinė problema, su kuria susiduriama siekiant apskaičiuoti transporto priemonių greitį panaudojant magnetinio lauko jutiklius yra mažas diskretizavimo dažnis. Dauguma pigių MEMS tipo magnetinio lauko jutiklių šiai dienai gali pasiekti 1kHz diskretizavimo dažnį, o tai yra nepakankama, norint tiksliai apskaičiuoti greitai važiuojančių transporto priemonių greitį.

Pagrindinis šio darbo tikslas – ištirti įvairius vėlinimo tarp dviejų signalų radimo būdus, įvertinti jų privalumus bei trūkumus ir pasiūlyti naują algoritmą, kuriuo būtų galima apskaičiuoti

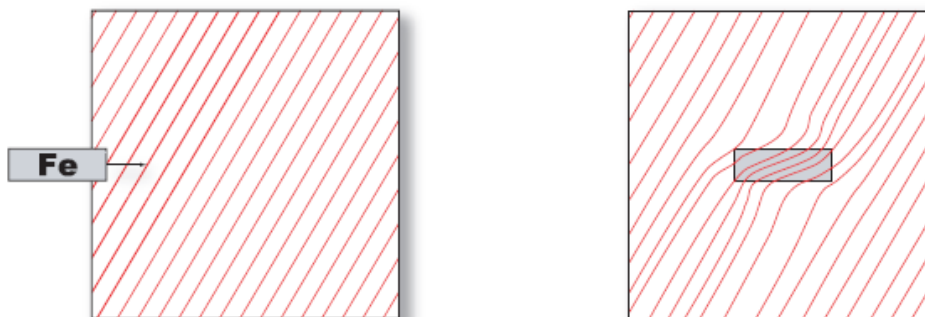
pravažiuojančios transporto priemonės greitį. Skaičiavimai turėtų būti atlikti kaip įmanoma greičiau, o greitis būtų apskaičiuojamas neviršijant užsiduotos paklaidos. Šiam darbui keliami uždaviniai:

- Išanalizuoti magnetinio lauko jutiklių panaudojimo galimybes transporto srautų valdyme.
- Išanalizuoti transporto priemonių aptikimo ir greičio nustatymo algoritmus, panaudojant magnetinio lauko jutiklių duomenis.
- Ištirti filtravimo įtaką transporto priemonių greičio nustatymo tikslumui
- Surasti racionaliausią transporto priemonių greičio nustatymo algoritmą ir ištirti jį eksperimentiškai.

1. LITERATŪROS ANALIZĖ

Šioje dalyje apžvelgiama literatūra ir straipsniai, analizuojama, kokie metodai yra taikomi transporto priemonių greičiui nustatyti panaudojant magnetinio lauko jutiklius. Daugelyje šaltinių analizuojami metodai, kaip atpažinti ir klasifikuoti transporto priemones pagal magnetines signatūras, kurias palieka transporto priemonė pravažiavusi virš dviejų magnetinio lauko jutiklių.

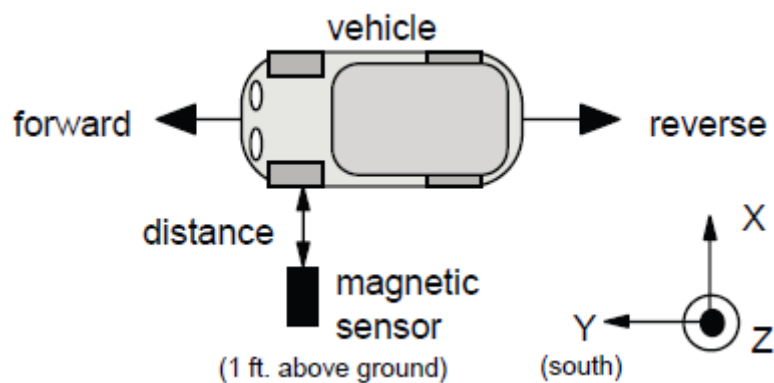
Žemė yra kaip didelis magnetas, o kiekvienas objektas esantis magneto įtakos zonoje iškreipia magnetinį lauką [3]. Žemės magnetinis laukas, yra nepastovus ir nuolatos keičiasi. Transporto priemonių aptikimui panaudojant magnetinio lauko jutiklius yra labai svarbu, kad Žemės magnetinis laukas nesikeistų per greitai tuo metu, kai pravažiuoja transporto priemonė. Be to, naudojant tuos pačius magnetinio lauko jutiklius gautume skirtingas amplitudes iš magnetinio lauko sensorių, taigi norint sistemą pritaikyti visame pasaulyje reikėtų tam tikro jos kalibravimo ir nuolatinio sistemos sekimo.



1 pav. Kairėje - žemės magnetinio lauko linijos kai transporto priemonės tam tikroje zonoje nėra, dešinėje - kai toje zonoje juda transporto priemonė [4]

Iš 1 pav. matyti, kad transporto priemonės judėdamos Žemės magnetiniame lauke iškreipia magnetinio lauko linijas, kad ir kur jos bevažiuotų, vadinasi šį magnetinio lauko iškrypimą galima aptikti panaudojant magnetinio lauko jutiklius.

Pagrindinė problema naudojant magnetinio lauko jutiklius yra ta, kad norint su jais išmatuoti transporto priemonės greitį yra pageidautina, kad transporto priemonės centrinė ašis būtų virš magnetinio lauko jutiklio. Žinoma transporto priemonė gali ir pravažiuoti tam tikru atstumu nuo jutiklio ir ji vis tiek iškreips magnetinį lauką, tačiau problema yra ta, kad bet kokiame atveju jutiklis turėtų būti sumontuotas po asfaltu, o tai daugeliu atveju yra nepriimtina dėl to, kad reikalinga gremžti važiuojamąją dangą. Apie tai plačiau kalbama šaltinyje [5], kad magnetinio lauko jutiklis idealiai turėtų būti sumontuotas šalia važiuojamosios kelio dalies (2 pav.), kad jos nereikėtų gadinti.



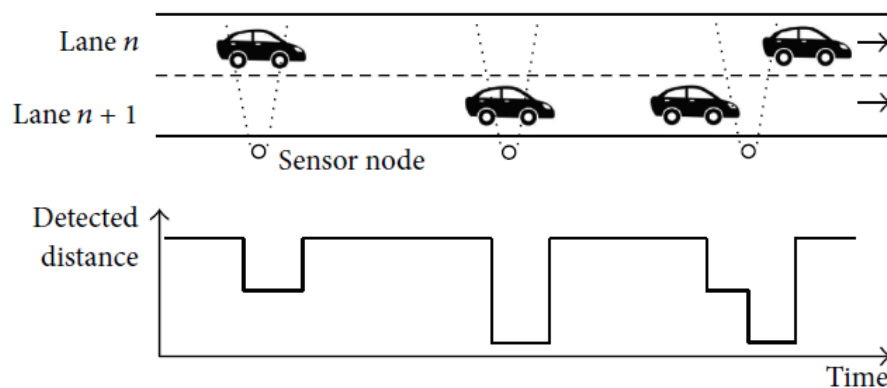
2 pav. Transporto priemonės ir jutiklio orientacija atliekant bandymus [5]

Iš anksčiau pateikto modelio matome, kad transporto priemonė pravažiuoja tam tikrame aukštyje nuo magnetinio lauko jutiklio, o magnetinio lauko jutiklis orientuojamas taip, kad jutiklio Z ašis būtų statmena į automobilio dugną. Tokiu būdu gaunamas didžiausias magnetinio lauko linijų iškraipymas šioje ašyje.

1.1 TP aptikimas

Literatūroje galima rasti daug metodų skirtų transporto priemonėms aptikti. Pagrindinė problema bandant aptikti TP yra ta, kad jų yra daug skirtingų rūšių – lengvieji automobiliai, sunkvežimiai, autobusai, mikroautobusai, motociklai ir t.t. Dėl šios priežasties nėra vieno TP transporto aptikimo metodo kuris tiktų visoms TP rūšims. Vieni metodai geriau aptinka vienos rūšies TP, tačiau neaptinka kitų rūšių. Dar viena problema, kodėl nėra vienintelio TP aptikimo metodo yra ta, kad TP aptikti reikia skirtingose geografinėse vietose. Kai TP juda užmiestio keliuose, kuriuose leidžiamas didelis greitis, o mieste jis mažas, todėl tas pats metodas gali nebetikti, o sankryžose ar stovėjimo aikštelėse reikia aptikti stovinčias TP. Žemiau pateikiamas trumpa pagrindinių metodų sątrauka, kurie naudojami TP aptikimui.

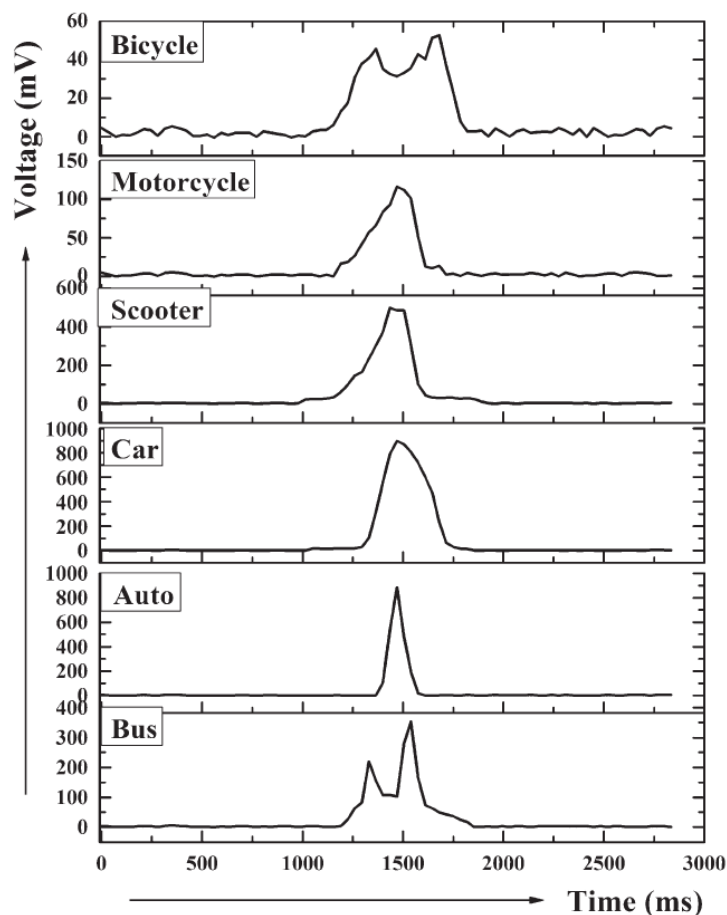
- Vienas iš būdų kaip galima aptikti TP yra panaudojant ultragarsinius keitiklius. Ultragarsiniai metodai TP aptikti stovėjimo aikštelėse naudojami jau gana seniai, tačiau iki šiol nebuvo labai daug bandymų pritaikyti ultragarsinius keitiklius judančioms transporto priemonėms keliuose aptikti. Iš principo šis metodas yra labai paprastas, tačiau kaip ir kiti metodai reikalauja signalų apdorojimo. Keitiklis išspinduliuoja impulsų seką, o po to ją bando užfiksuoti, atsispindėjusią nuo tam tikros kliūties. Jeigu signalas atgal nesugrįžta, vadinasi siųsto signalo kelyje nebuvo TP, o jeigu signalas grįžo, vadinasi tuo metu pro jį judėjo kokio nors tipo TP. Pavyzdžiui [17] nagrinėjamas metodas TP aptikti dvejose juostose. Ultragarsinis keitiklis pastatomas šalia važiuojamosios kelio dalies. Pagal tai, kiek laiko praėjo kol signalas grįžo nuo atsispindėjusios kliūties galima spręsti, kurioje juostoje važiavo transporto priemonė.



3 pav. TP aptikimas ultragarsiniais keitikliais [17]

Pagrindinis tokių sistemų trūkumas yra tas, kad jeigu dvi transporto priemonės juda lygiagrečiai, tai niekaip neįmanoma jų atskirti. Sistemai atrodo, kad pravažiavo viena ilga transporto priemonė. Dar viena problema su kuria susiduria ši sistema yra ta, jog TP keičiant savo judėjimo juostą sistema gali užregistruoti ją kaip naują TP, nors iš tikrųjų TP tiesiog keitė važiavimo juostą.

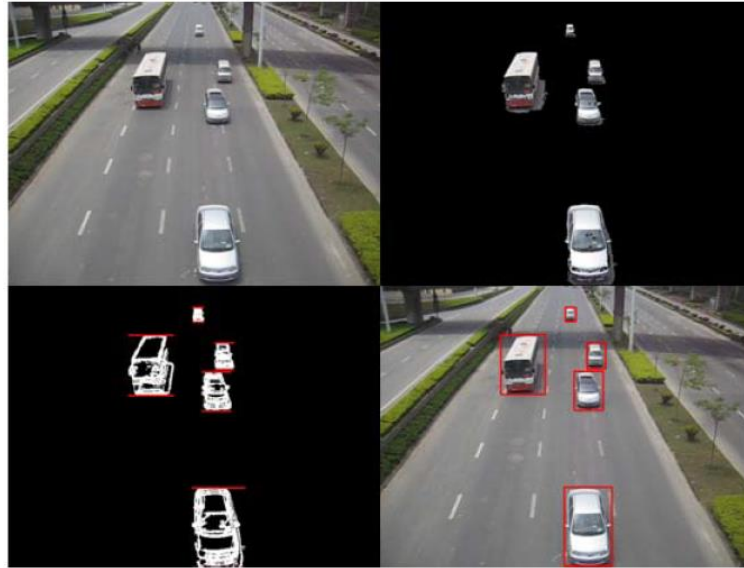
- Turbūt vienas seniausių metodų TP aptikti aprašytų literatūroje yra indukcinis metodas. Jis pagrįstas tuo, kad važiuodama TP pro ritę, kuria teka elektros srovė, iškraipo tos ritės kuriamą magnetinį lauką. Lauko iškraipymai atsiranda, nes TP juda kintamame magnetiniame lauke ir joje indukuojasi sukūrinės srovės metalo paviršiuose, kurios kuria savo magnetinį lauką. Tuomet ant kelio įtaisytos ritės ir TP sukurtas magnetinis laukas susisumuoja ir galime stebėti magnetinio lauko pokyčius. Pavyzdžiui [18, 19] nagrinėjamos metodikos, kaip indukcinio kilpų dėka ne tik aptinkamos TP, bet dar ir klasifikuojamos pagal magnetinio lauko pokyčius.



4 pav. TP aptikimas ir klasifikavimas naudojant indukcinės kilpas [18]

Plačiau TP klasifikavimą aptarsime skyrelyje „TP klasifikavimas“.

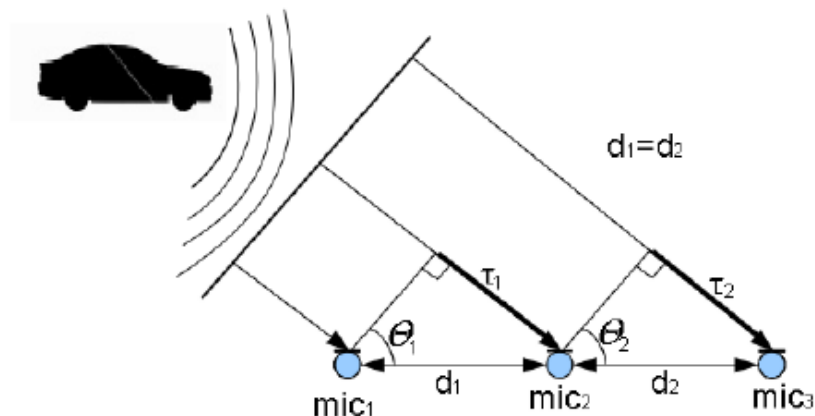
- TP aptikimas vaizdo kameromis yra ganėtinai senas metodas. Galima rasti ne vieną straipsnį šia tematika, o TP aptikimo algoritmai nelabai kuo skiriasi. Pavyzdžiui [13], [14] TP aptikimui naudojama metodika, kuomet pirmiausia parenkamas fonas, kuriame būtų lengva aptikti objektus. Dažniausiai yra parenkamas juodas fonas. Tuomet padaroma nuotrauka, su kuria bus lyginami visi kiti kameros kadrai. Kai į kadrą papuola kokio nors tipo TP, panaudojant tam tikrus algoritmus ir palyginus vaizdą su etaloniniu, galima išskirti TP kontūrus ir jas pažymėti.



5 pav. TP aptikimas su vaizdo kamera pritaikius vaizdų apdorojimo algoritmus [13]

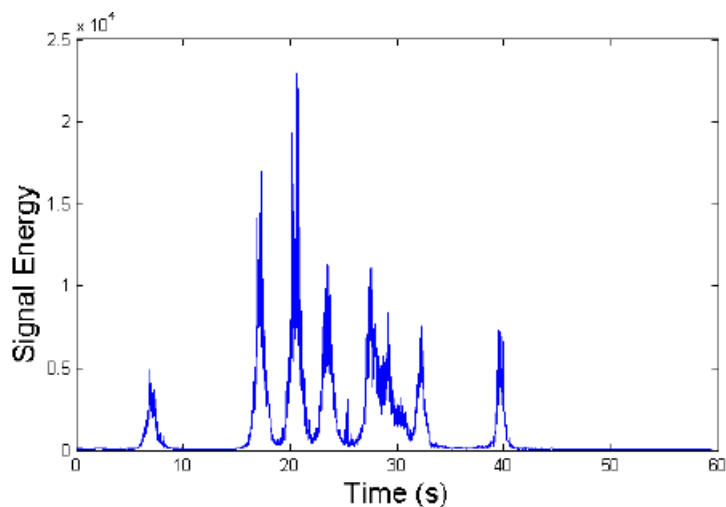
Visi TP aptikimo metodai naudojant vaizdo kameras reikalauja daug resursų – reikalingos brangios kameros, programinė įranga vaizdams apdoroti, kompiuteriai, galingi mikrovaldikliai arba FPGA matricos, kurios pajėgtų apdoroti didelius kiekius duomenų labai greitai, nes TP aptikimas turi vykti realiu laiku.

- Vienas iš įdomesnių TP aptikimo metodų yra panaudojant mikrofonus. Literatūroje galima rasti straipsnių šia tematika, tačiau niekur nenaudojamas vienas mikrofonas, dažniausiai naudojamas mikrofonų masyvas (keletas ar keliolika). Pavyzdžiui [15], [16] toks mikrofonų masyvas išbandomas praktiškai. TP artėjant prie masyvo, pagal signalo vėlinimą tarp mikrofonų galima apskaičiuoti kampą tarp TP ir masyvo, o pagal tai galima skaičiuoti pravažiuojančias TP. Taip pat paskaičiavus pirmą išvestinę galima nustatyti TP judėjimo kryptį (6 pav.).



6 pav. TP aptikimas mikrofonų masyvu [16]

Taip pat TP galima aptikti pagal mikrofonais priimto signalo energiją.



7 pav. Vienos minutės mikrofonų įrašas, pro kuriuos judėjo TP [16]

Šis metodas turi keletą trūkumų. Pavyzdžiui ta pati TP pravažiavus didesniu atstumu nuo mikrofonų masyvo sukeltų ženkliai mažesnės amplitudės signalus, dėl ko būtų sudėtingiau juos analizuoti. Taip pat tyrimų metu pastebėta, kad šiuo metodu ganėtinai sunku aptikti TP kai oras labai vėjuotas, arba ganėtinai stipriai lyja. Tuomet mikrofonuose sukuriama didelės amplitudės triukšmai, dėl ko aptikti TP pasidaro sudėtinga ir metodai, kurie aptarti šiuose straipsniuose, nebeduoda tokių gerų rezultatų.

1.1.1 TP aptikimas AMR tipo magnetinio lauko jutikliais

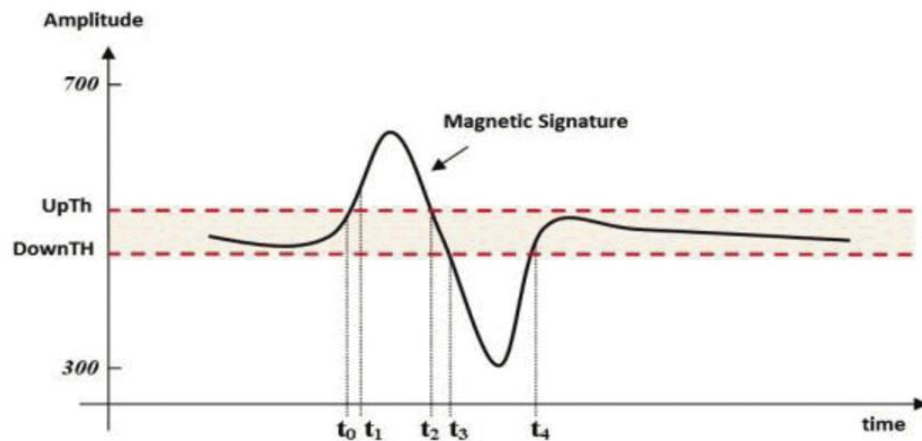
Vienas iš būdų, kaip galima aptikti TP tai panaudoti magnetinio lauko jutiklius ir užfiksuoti magnetinio lauko iškreipimus, kurie gaunami, kai TP pravažiuoja virš magnetinio lauko jutiklių. Pagrindinė problema su kuria susiduriama taikant šį metodą yra ta, kad TP turi labai daug dalių, kurios gali sukelti ne vieną svyravimą, kol TP visiškai pravažiuos pro jutiklius. Ši problema sprendžiama užfiksuojant TP tik tada, kai magnetinėje signatūroje viršijamos tam tikros užduotos slenkstinės reikšmės.

Yra ne vienas būdas, kaip galima nustatyti šias slenkstines vertes, kurias kirtus konstatuojamas faktas, jog pravažiavo transporto priemonė. Pavyzdžiui gali būti naudojamos dvi slenkstinės linijos, kurios nustatomos pagal formules [12]:

$$UpTh = base(k) + \delta \quad (1)$$

$$DownTh = base(k) - \delta \quad (2)$$

čia δ - konstanta, parenkama empiriškai. Šiuo modelių paremtas TP aptikimo būdas parodytas 8-ame paveiksle:



8 pav. Pavyzdinė signatūra, kai TP juda virš magnetinio lauko jutiklio [12]

TP aptikti šio straipsnio autoriai naudoja būsenų mašiną (angl. *State machine*), kurioje yra 9-ios būsenos.

Case	V ₀	V ₁	V ₂	Action	Signal Form
1	L0	L0	L0	No action	UPTh DownTh
2	L0	L0	L+	Report a detection	
3	L0	L+	L0	CO = CO+1 if (CO = 2) CO = 0; V1 = L0 End if	
4	L+	L0	L+	Report a detection CO = 0	
5	L+	L0	L-	No action	
6	L0	L-	L0	CO = CO +1 if (CO = 2) CO = 0 ; V1 = L0 End if	
7	L-	L0	L+	No action	
8	L0	L0	L-	Report a detection	
9	L-	L0	L-	Report a detection CO=0	

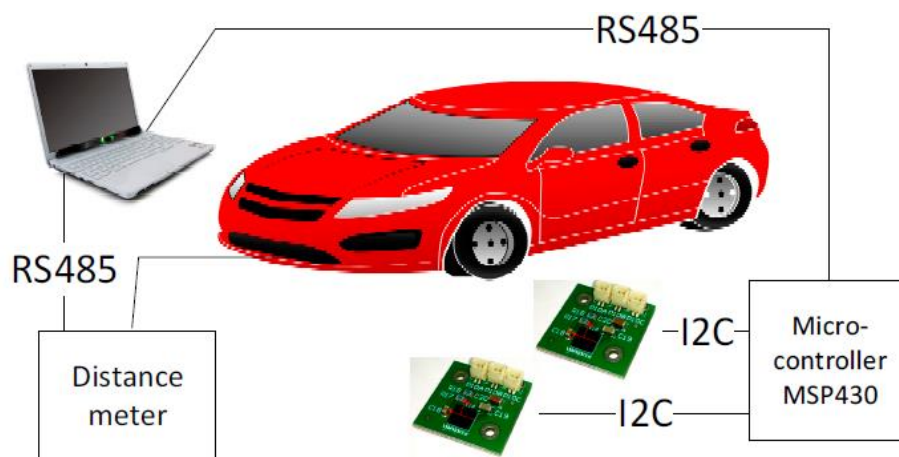
9 pav. Būsenų mašina naudojama TP aptikti [12]

Anksčiau pateiktame paveiksle L0 reiškia magnetinio lauko jutiklio būseną, kai jutiklio duomenys yra tarp *UpTh* ir *DownTh*. L+ užfiksuojama kai jutiklio duomenys yra virš *UpTh*, o L- užfiksuojama kai jutiklio duomenys yra žemiau *DownTh*.

Naudojant 5 pav. pateiktą pavyzdį, galima aptarti, kaip šios būsenų mašinos dėka yra aptinkamos TP. Skaitiklis CO naudojamas skaičiuoti svyravimų skaičiui signatūroje. Būsena Nr. 1 yra pradinė būsena. Šioje būsenoje jutiklis būna po kalibracijos, arba kai jutiklis yra ramybės būsenoje ir nėra

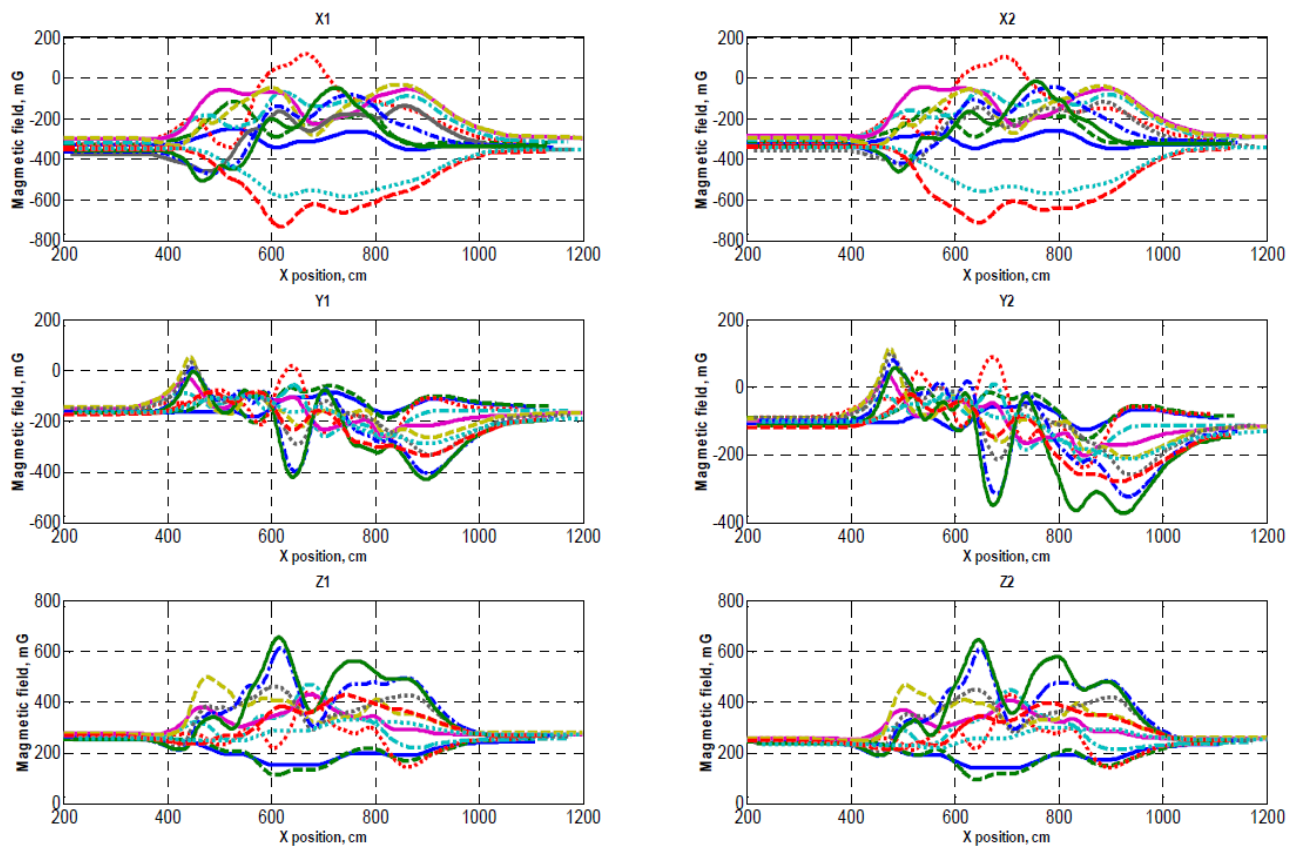
jokių išorinių veiksnių, kurie galėtų pakeisti magnetinio jutiklio duomenis. Būsenos 2 ir 8 yra TP fiksavimo pradžios momentas. Būsenos 3 ir 6 skirtos aptikti pirmojo svyravimo pabaigą. Skaitiklis CO rodo, kuris svyravimas tai buvo, pirmasis ar antrasis. Vienai TP fiksuoti naudojami du susvyravimai. Jeigu aptinkamas antrasis susvyravimas, TP aptikimas baigiamas ir galima konstatuoti, kad virš jutiklio pravažiavo TP, apie kurią galima pranešti. Būsenos 4 ir 9 skirtos atpažinti naujo svyravimo pradžią, kai prieš tai buvęs svyravimas buvo į tą pačią pusę. Būsenos 5 ir 7 skirtos atpažinti naujo svyravimo pradžią, kai prieš tai buvęs svyravimas buvo į priešingą pusę. Jeigu patenkame į 5 arba 7 būsenas, tai reiškia, kad pravažiavusi virš jutiklio TP sukėlė ne vieną, o keletą svyravimų [12].

Straipsnyje [20] aptariama naujo tipo TP aptikimo sistema panaudojant AMR tipo magnetinio lauko jutiklius. Sistemos struktūra pavaizduota 10 pav.



10 pav. Eksperimento su dviem AMR tipo magnetinio lauko jutikliais schema [20]

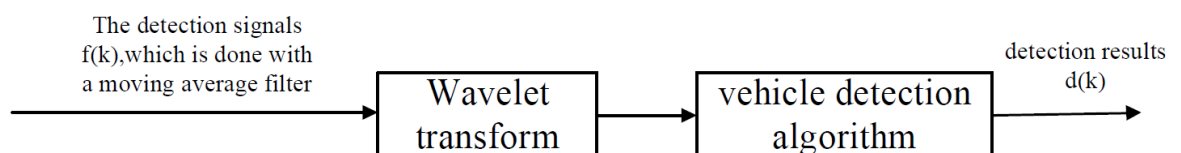
Sistemos struktūrą sudaro du AMR tipo jutikliai, iš kurių duomenis per I2C sąsają priima ir apdoroja mikrokontroleris MSP430. Atstumo matuoklis naudojamas fiksuoti magnetinio lauko komponentes kas 1cm. Šis eksperimentas buvo atliekamas siekiant nustatyti TP aptikimo galimybes stovėjimo aikštelėse. Apdorojus eksperimentų matavimo rezultatus autoriai parodė, kad magnetinio lauko iškreipimus aplink AMR magnetinio lauko jutiklį labai priklauso nuo pozicijos, kaip pro jutiklį pravažiuoja TP. Iš čia autoriai padarė labai svarbią išvadą, kad norint aptikti TP ir mėginti užfiksuoti jų greitį naudojant magnetinio lauko jutiklius, reikia išmatuoti visas tris magnetinio lauko komponentes ir tik panaudojant visas jas galima gauti kriterijų, kuris aiškiai apibrėš TP aptikimo sąlygas. Atskirų komponentių nagrinėjimas šiuo atveju neturi prasmės. Žemiau pateikiami gauti eksperimentų rezultatai, kaip magnetinis laukas yra iškreipomas aplink du sensorius kai virš jų juda TP. Skirtingomis spalvomis šiuose grafikuose paženklintos skirtingos TP.



11 pav. Magnetinio lauko iškreipimas aplink abudu jutklius virš jų judant skirtingo tipo TP [20]

Straipsnyje taip pat aptariami metodai TP greičio apskaičiavimui ir algoritmai tikslesniam TP aptikimui, tačiau šie metodai atsiremia į signalų apdorojimą, todėl plačiau apie šiuos metodus šiame skyrelyje nekalbėsime.

Dar viena metodika, kurią galima naudoti TP aptikimui, naudojant magnetinio lauko jutklius, yra Vilnelių transformacija (angl. Wavelet transform) [21]. Pagrindiniai tokio metodo privalumai yra didelės signalų apdorojimo galimybės, be to užfiksuojama signalo informacija dažnio srityje. Taip pat galimybės signalų suspaudimui ir triukšmo pašalinimui, kas leidžia gauti geros kokybės signalus. Tačiau šis metodas yra netinkamas realaus laiko sistemoms dėl ilgos skaičiavimo trukmės realiuose mikrovaldikliuose. Žemiau pateikiamas šio metodo pritaikymo algoritmas:

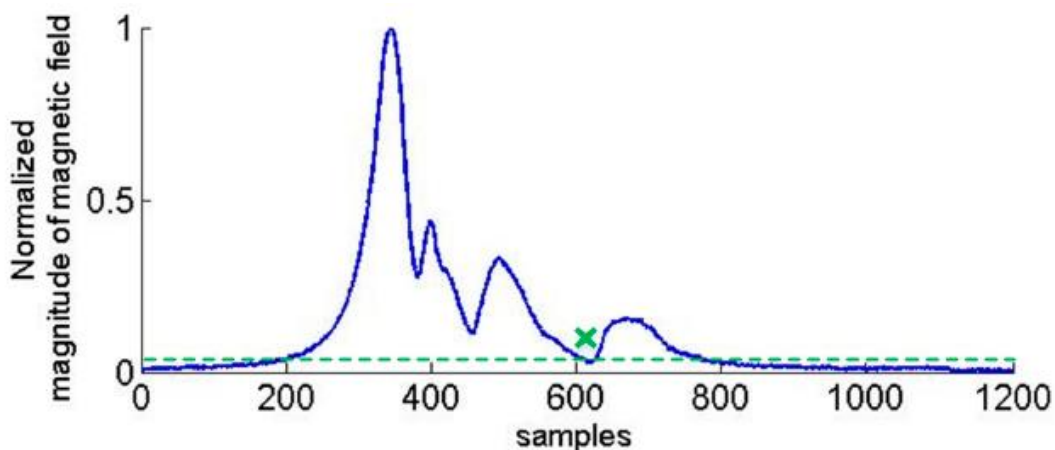


12 pav. TP aptikimo sistemos architektūra [21]

Taip pat šiame straipsnyje aptariamos TP greičio nustatymo galimybės ir metodikos panaudojant tuos pačios magnetinio lauko jutiklio duomenis, tačiau apie juos plačiau bus kalbama poskyryje „TP greičio nustatymas“.

Dar daugiau TP aptikimo metodų analizuojama straipsnyje [22]. Taip pat šiame straipsnyje analizuojamos TP greičio bei ilgio nustatymo galimybės bei metodai. Taip pat šiame straipsnyje analizuojama galimybė nustatyti transporto priemonės ilgį pagal magnetinio lauko jutiklių duomenis. Trumpai aptarsime kiekvieną šiame straipsnyje nagrinėjamą TP aptikimo metodą:

- Pirmasis metodas nesiskiria niekuo nuo daugelyje straipsnių aptariamo metodo – tai paprastas TP fiksavimas, kai magnetinė signalūra viršija tam tikra slenkstinę reikšmę. Slenkstinė reikšmė parenkama empiriškai. Pagrindinė šio metodo problema yra ta, kad skirtingo tipo TP turi labai skirtingų amplitudžių signalūras, todėl tokiu būdu nustatyta slenkstinė vertė gerai tiks aptikti vieno tipo TP, tačiau visiškai gali netikti aptikti kito tipo TP. Taip pat jeigu slenkstinė vertė per maža, gali būti jog užfiksuosime signalūros pabaigą anksčiau nei ji iš tikrųjų įvyko, kaip tai parodyta žemiau esančiame paveiksle:

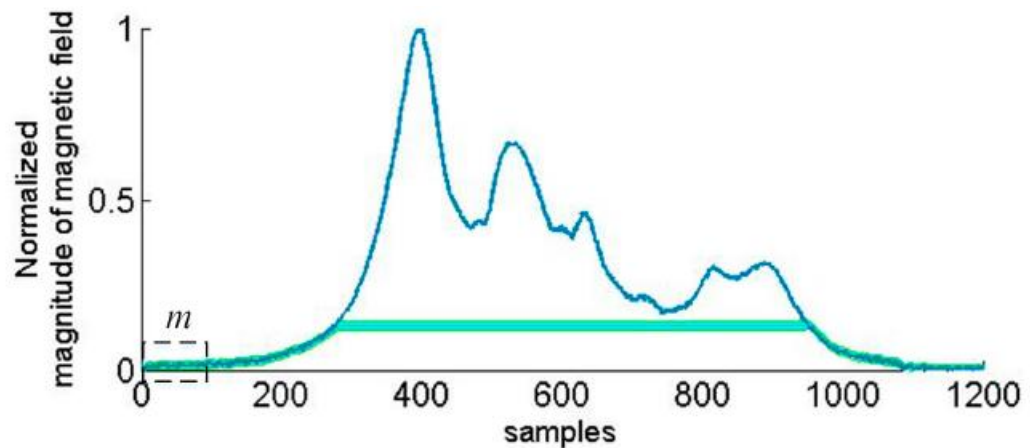


13 pav. TP aptikimas kai magnetinė signalūra viršija slenkstinę vertę (klaidingas signalūros pabaigos aptikimas) [22]

- Antrajame metode TP taip pat aptinkama kai kertama slenkstinė vertė, tačiau šiuo atveju slenkstinė vertė nustatoma suskaičiavus pirmųjų m reikšmių slenkantį vidurkį. Suskaičiavus slenkantį vidurkį, apskaičiuojama standartinė deviacija. Nustatoma slenkstinė vertė yra lygi pirmųjų m atskaitų vidurkio ir k kartų standartinės deviacijos sumai:

$$SL = SMA(m) + k * stdev. \quad (3)$$

. čia SL – slenkstinė reikšmė. Skaitiklis k yra svarbus, nes jeigu taško vertė yra k kartų standartinių deviacijų toliau nei slenkamasis vidurkis, tuomet užfiksuojamas pikas. Šie pikai vėliau naudojami TP ilgiui nustatyti. Plačiau apie tai skyrelyje „TP klasifikavimas“. Žemiau pateikiamas metodo esmę iliustruojantis paveikslas:



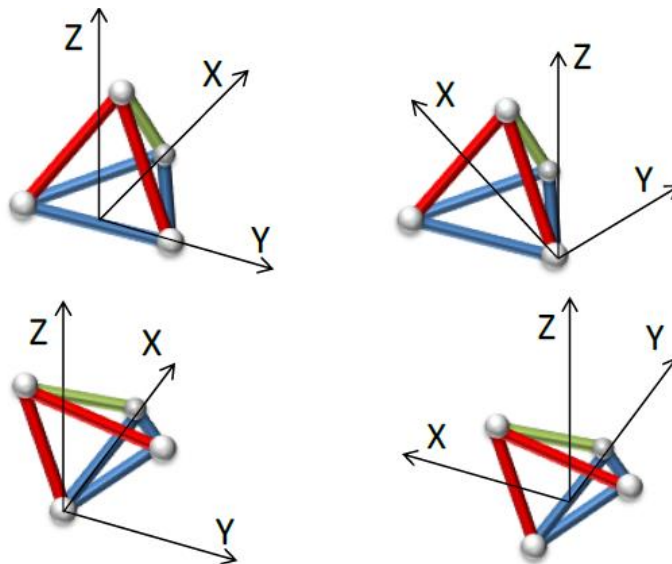
14 pav. TP aptikimas kirtus slenkstinę vertę, kai ši nustatoma pagal slenkantį vidurkį ir standartinę deviaciją [22]

Kiti šiame straipsnyje aptariami TP aptikimo metodai yra labiau skirti ne bendram TP aptikimui, o TP klasifikavimui pagal ilgį, todėl plačiau šie metodai bus aptarti kitame skyrelyje.

1.2 TP klasifikavimas

Klasifikuoti TP galima panaudojant daug skirtingų metodų, kurių populiariausi yra: indukcinės kilpos, vaizdinis aptikimas naudojant kamerą, klasifikavimas naudojant ultragarsinius jutiklius arba mikrobangų jutiklius, bei pjezoelektrinius jutiklius [11]. Du populiariausi TP klasifikavimo kriterijai yra TP klasifikavimas pagal ilgį ir TP klasifikavimas pagal modelį (krovinis automobilis, lengvasis automobilis, autobusas, mikroautobusas ir t.t.).

Transporto priemonių klasifikacija atliekama analizuojant magnetines signatūras, kurios gaunamos kai TP pravažiuoja virš magnetinio lauko jutiklio. TP aptikimui tokiu atveju naudojamas vienas magnetinis jutiklis arba dviejų jutiklių pora. Kad tokiu būdu įmanoma klasifikuoti TP yra daug straipsnių. Straipsnyje [10] autoriai pateikia TP aptikimo matematinį modelį naudojant AMR tipo magnetinio lauko jutiklius. Modelio adekvatumo patikrinimui autoriai naudojo trijų ašių magnetinio lauko jutiklį „Honeywell HMR2300“. Sudaryta magnetinė tetraedrų struktūra, nuo kurios posvyrio kampo priklauso magnetinio lauko jutiklio duomenys.

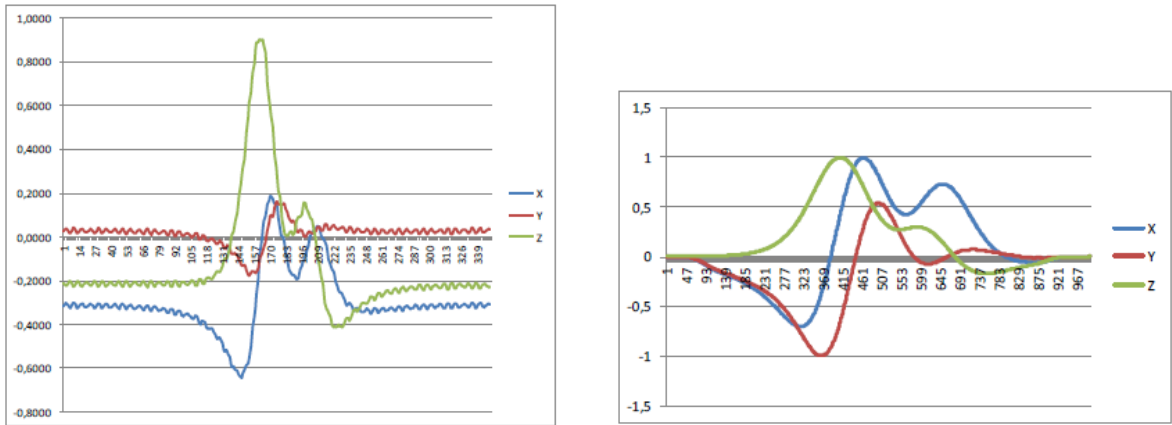


15 pav. Keturi pagrindinės tetraedro kryptis į skirtingas ašis [10]

Gaunami signalai tokiai struktūrai keliaujant virš magnetinio lauko jutiklio dažniausiai yra tolydūs ir nesudėtingi. . Pagrindiniai keturi žingsniai apdorojant gautus signalus yra [10]:

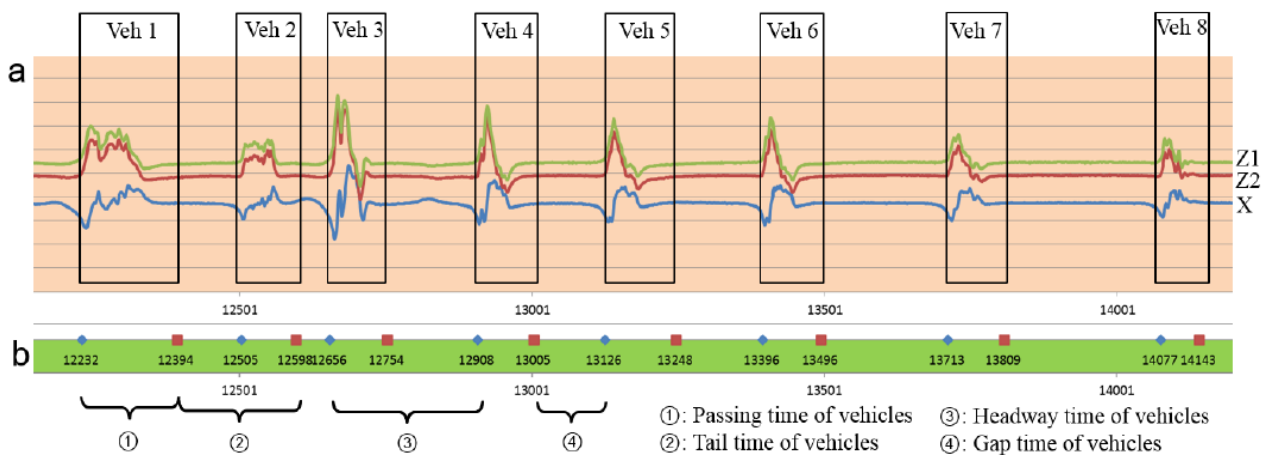
- Triukšmo iš signalo pašalinimas panaudojant ŽDF.
- Tikslus signalo pradžios ir signalo pabaigos pasirinkimas pagal pasirinktus kriterijus.
- Signalų perdiskretizavimas, kad gauti signalą su 1000 atskaitų.
- Signalų normalizavimas x,y ir z ašyse, kad visose signalo ribose būtų nuo -1 iki +1.

Signalų normalizavimas reikalingas tam, kad galėtume palyginti įvairius skirtingus signalus, nes skirtingos TP turės skirtingų ilgių ir skirtingų amplitudžių signalūras, todėl be signalų amplitudžių normalizavimo neįmanoma palyginti skirtingų TP signalūrų. Tuo tarpu dėl trečiojo punkto, kuris teigia jog reikia signalą perdiskretizuoti taip, kad gautume signalą su 1000 atskaitų galima nesutikti. Argumentai kodėl tai nėra būtina pateikiami toliau nagrinėjamuose šaltiniuose, tačiau pagrindinė priežastis yra ta, jog skirtingu greičiu važiuojančios TP turės skirtingo ilgio signalūras, todėl signalūrų ilgius galima suvienodinti tik tuo atveju, jeigu žinoma, kad skirtingos TP pravažiavo pro magnetinio lauko jutiklį vienodu greičiu. Žemiau pateikiamas eksperimentų metu gautos signalūros pavyzdys ir apdorotas signalų vaizdas.



16 pav. Gauti signalai eksperimento metu (kairėje) ir apdoroti signalai (dešinėje) [10].

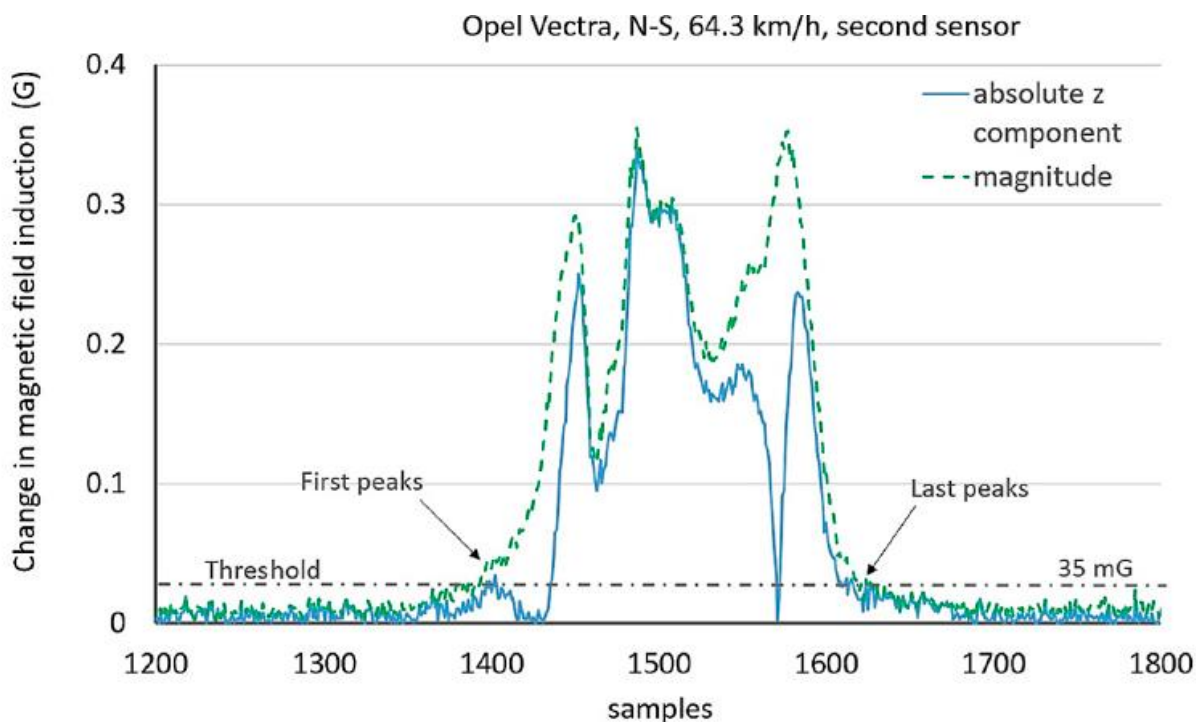
Iš daugelio pravažiuojančių transporto priemonių galima surinkti statistiką ir stebėti kaip jos iškraipo magnetinį lauką bei klasifikuoti jas pagal modelį ar ilgį, nes kiekviena transporto priemonė turi gan skirtingą magnetinės signalo statumą, formą, amplitudę [7]. Iš žemiau pateikto paveikslo matyti, kad skiriasi visi aukščiau išvardinti parametrai. Skirtingas signalų ilgis taip pat reiškia, kad taikant įvairius metodus transporto priemonės greičio apskaičiavimui, kiekvienas atvejis bus individualus laikiniu požiūriu, taigi tokia sistema niekada neturės tikslaus apibrėžimo, kiek užtrunka apskaičiuoti transporto priemonės greitį, nes tai tiesiogiai susiję su signalo ilgiu.



17 pav. Įvairios transporto priemonės skirtingai iškraipo magnetinį lauką [7]

Ankstesniame skyrelyje aptarėme straipsnyje [22] analizuojamus TP aptikimo metodus naudojant slenkstinę vertę. Toliau aptariame to paties straipsnio metodus, kurie naudojami TP detekcijai, tačiau šių metodų galutinis tikslas yra ne aptikti TP, o nustatyti TP ilgį.

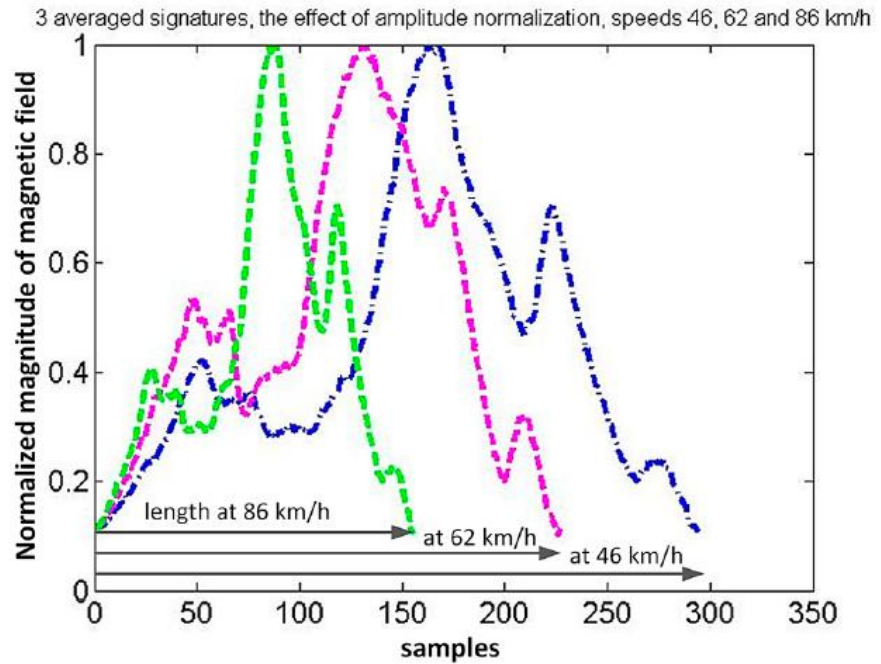
- Pirmasis metodas yra dviejų kraštinių pikų metodas. Apskritai aptikti TP naudojamas bet koks slenkstinės vertės kirtimo metodas, tačiau tokiais metodais tiksliai neįmanoma nustatyti TP ilgio. Vienas iš būdų yra ieškoti dviejų kraštinių pikų, kurie tampa atskaitos taškais TP ilgio skaičiavimui. Paprastai gautoje magnetinė signaloje tokių pikų būna ne vienas ir ne du. Žemiau pateiktame paveiksle matome šio metodo iliustraciją:



18 pav. TP magnetinėje signalūroje užfiksuoti šoniniai pikai [22]

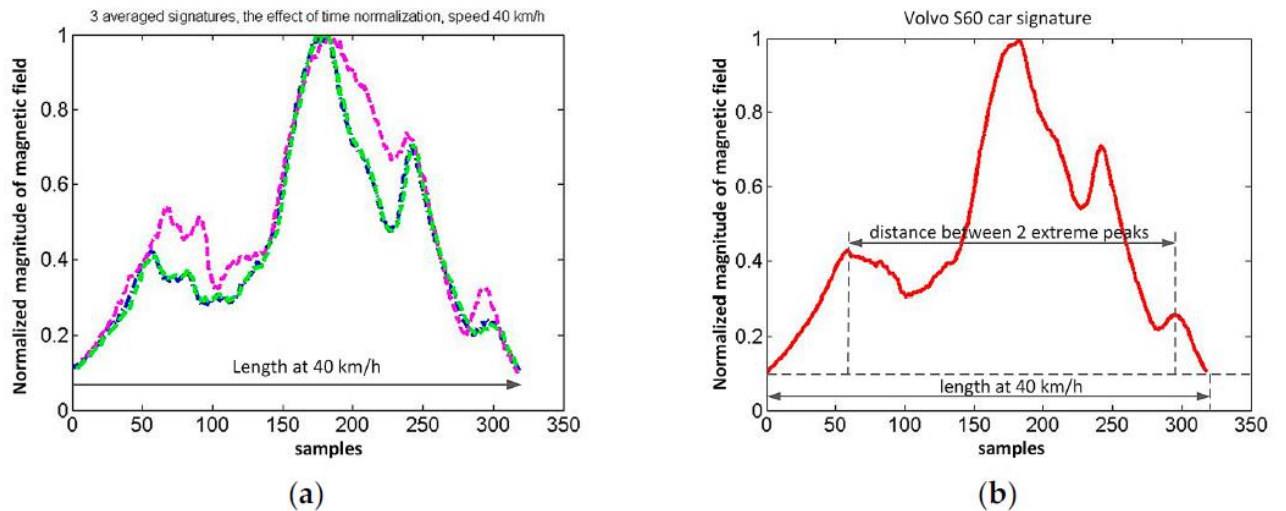
Pagrindinis šio metodo trūkumas yra tas, kad apie slenkstinę vertę gali susiformuoti ne vienas jos kirtimas, dėl to tiksliai sunku nustatyti kuri reikšmė yra teisingiausia. Šiai problemai spręsti įvedamas kintamasis, kuris apsprendžia minimalų atstumą tarp dviejų rastų pikinių reikšmių. Atstumas nuo vieno iki kito piko negali būti mažesnis nei ši reikšmė.

- Paskutinis straipsnyje aptariamas metodas yra skirtas TP klasifikacijai pagal TP tipą. Autoriai mano, jog kiekvienas TP modelis turi savo magnetinę signalūrą, kuri daugiau ar mažiau yra vienoda visiems to tipo modeliams. Metodas paremtas amplitudžių ir laiko normalizavimu pritaikant tiesinę interpoliaciją arba ekstrapoliaciją. Metodo esmė yra taikant slenkantį vidurkį iš n atskaitų, gauti tris signalų poras, kurios užfiksuotos tai pačiai TP važiuojant skirtingais greičiais. Tuomet normalizuojamos signalų amplitudės, o signalai laikinėje ašyje ekstrapolijuojami arba interpoliuojami, kad visos trys magnetinės signalūros būtų vienodo ilgio. Tuomet galutiniame rezultate išvedamas trijų signalūrų vidurkis ir gaunama tarsi etaloninė magnetinė signalūra, kuri gali būti saugoma duomenų bazėje. Tuomet kiekvieną kartą kai TP pravažiuos pro tokią sistemą ir bus užfiksuota jos magnetinė signalūra, panašios signalūros bus ieškoma duomenų bazėse. Jei atitikmuo bus rastas, tai reikš, kad tiksliai atpažintas transporto priemonės modelis. Žemiau pateikiama iliustracija, kokios magnetinės signalūros gaunamos, kai ta pati transporto priemonė tris kartus skirtingais greičiais pravažiuoja virš dviejų magnetinio lauko jutiklių.



19 pav. Normuotos amplitudės magnetinės signatūros TP judant skirtingais greičiais [22]

Žemiau pateiktame paveiksle matomas laiko ašies normalizavimo efektas.



20 pav. Laiko ašies normalizavimo efektas (a), trys magnetinės signatūros, gautos TP judant skirtingais greičiais. Atraminė greičio reikšmė yra 40 km/h. (b) matomas trijų signatūrų vidurkis. Galutinė signatūra turi du kraštinius pikus [22]

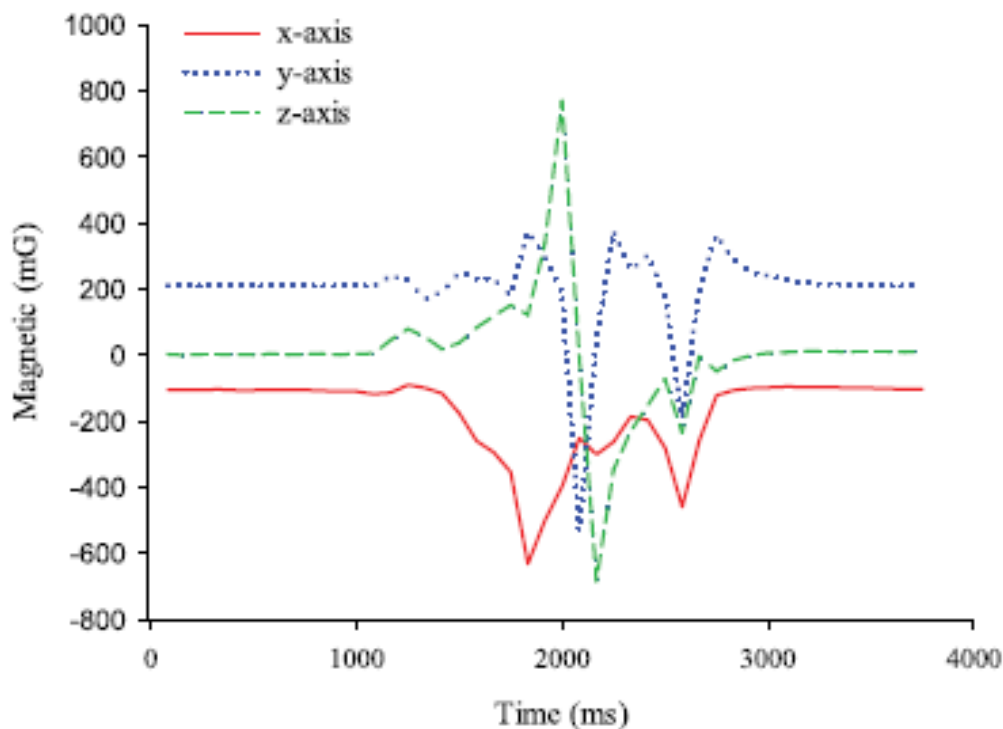
Autorių teigimu, pagal gautą galutinę magnetinę signatūrą galima bandyti kurti TP klasifikacijos sistemas pagal konkretų tipą, o gal net ir markę. Galutinė magnetinė signatūra gali būti saugoma kur nors duomenų bazėje, o bet kokiai transporto priemonei pravažiavus virš jutiklių, gauta signatūra būna lyginama su turimomis duomenų bazėje. Vienas klausimas, į kurį autoriai nepateikia atsakymo naudodami šį metodą yra tas, kodėl kaip etaloninis buvo pasirinktas 40 km/h greitis.

1.3 TP greičio nustatymas

Daugeliu atvejų iki šiol paprasčiausias metodas apskaičiuoti transporto priemonės greičiui naudojantis magnetinio lauko jutiklių duomenimis buvo naudojamas tik vienas magnetinis jutiklis [8]. Šis metodas duodavo prastus rezultatus. Modernesnis ir šiuolaikiškesnis metodas buvo aptartas [9]. Šiame metode jau buvo naudojami du magnetiniai jutikliai, kurie fiksuodavo magnetinio lauko pokyčius virš jų pravažiuojant transporto priemonei, o tuomet greitis iš gautos magnetinės signatūros buvo paskaičiuojamas paprasčiau nustatčius tam tikrą ribinę vertę ir užfiksavus, kuomet viena ir kita signatūra tą ribinę vertę viršija. Turint šį laiko skirtumą galima apskaičiuoti, koks buvo transporto priemonės greitis, nes atstumas tarp dviejų magnetinių jutiklių yra žinomas. Pagrindinis magnetinio lauko jutiklių privalumas prieš visas kitas metodikas, tokias kaip lazeriniai metodai, ultragarsiniai metodai, ar filmavimas kamera ir vaizdų apdorojimo atlikimas yra tas, kad magnetinio lauko jutiklis gali būti sumontuotas po kelio danga, vadinasi jis yra atsparus rūkui, drėgmei, lietai, šalčiui ir visiems kitiems aplinkos pokyčiams.

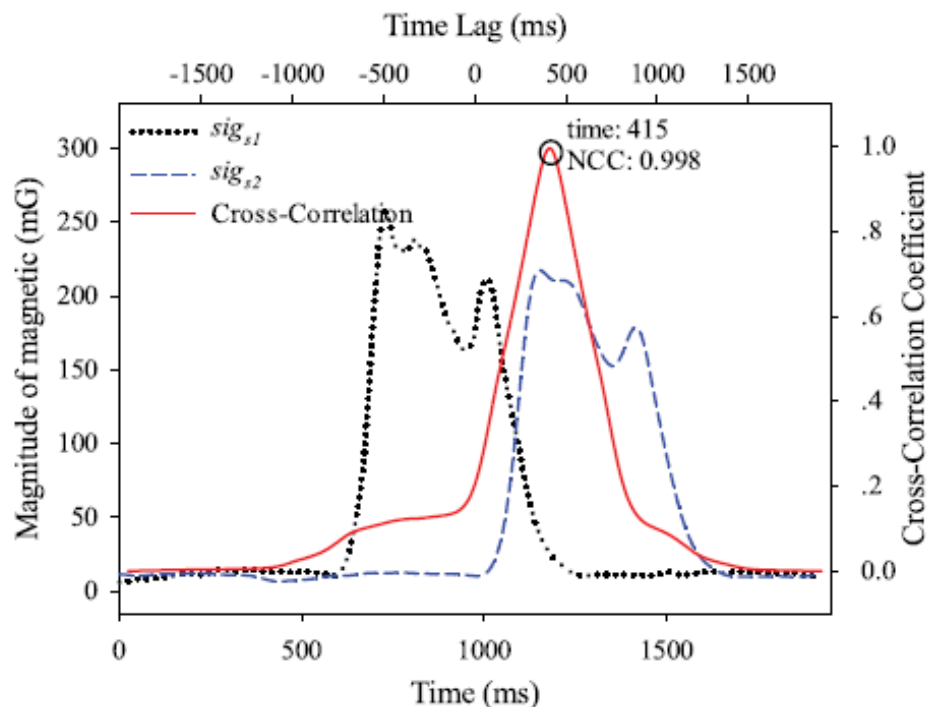
Nesenai buvo bandyta pritaikyti kros-koreliacijos metodą pritaikyti transporto priemonės aptikimui. Magnetinė signatūra, kuri gaunama transporto priemonei pravažius virš jutiklio, yra nufiltruojama slenkančio vidurkio filtro, kurio eilė yra parenkama eksperimentiškai, o kaip atraminis signalas parenkama Gauso funkcija, nes gautos magnetinės signatūros šlaitai labai panašūs į Gauso skirstinio [6]. Tačiau metodas labai sudėtingas, netikslus ir labai prastai veikia realioje aplinkoje, todėl vėliau šios idėjos buvo atsisakyta. Tame pačiame straipsnyje kalbama apie įvairias transporto priemonės greičio nustatymo metodikas, tačiau kros-koreliacija išskiriama kaip labai sudėtingas, energetiškai neefektyvus metodas tiek transporto priemonės aptikimui, tiek jos greičio nustatymui.

Žemiau pateikiamas vaizdas iš aptariamo straipsnio, kuriame aiškiai matosi, kaip pro jutiklius pravažiuojanti transporto priemonė iškraipo magnetinį lauką ir gaunamas įvairios signatūros kiekvieno jutiklio ašyje. Paprastai tolimesniam apdorojimui naudojamas visų ašių modulis, tačiau šiuo atveju tik norima parodyti veikimo principą. Taip pat matome, kad kaip ir minėta aukščiau, didžiausi iškraipymai matomi z ašyje nes ji yra Statmena transporto priemonės judėjimo kryptis.



21 pav. Magnetinio lauko jutiklio užfiksuotas iškraipymas kiekvieno ašyje virš jo judant transporto priemonei [6]

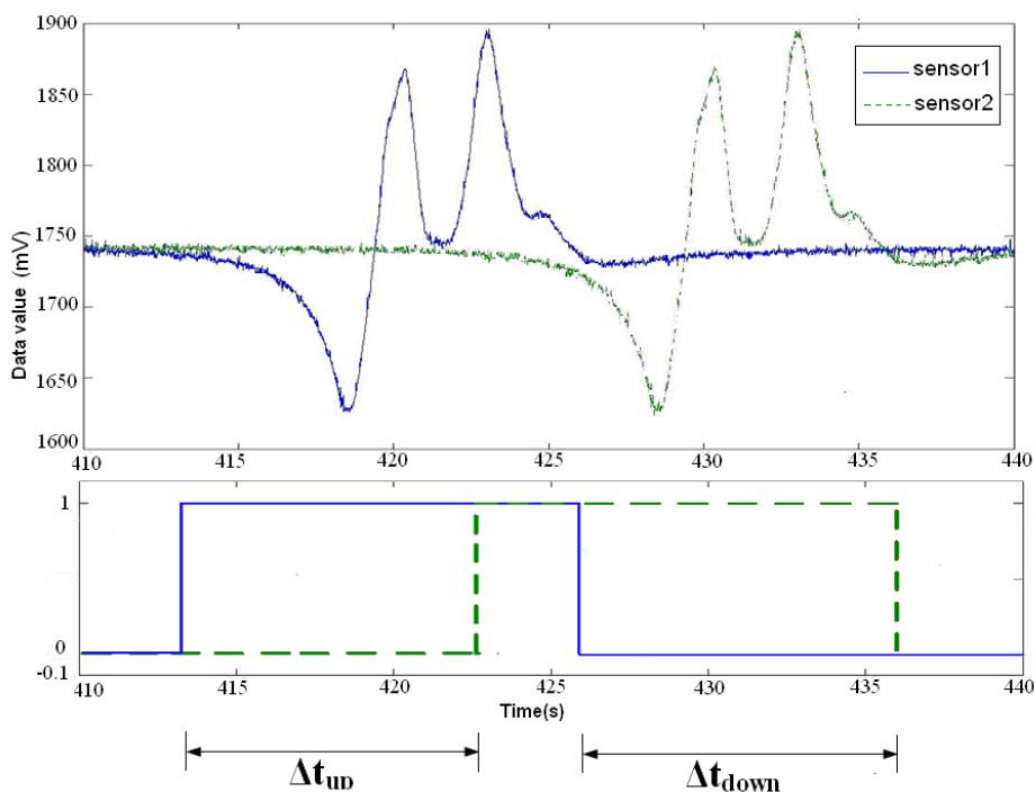
Tame pačiame straipsnyje pristatoma, kaip paėmus du magnetinio lauko jutiklius ir pastačius juos tam tikru atstumu vieną nuo kito, bei virš jų pravažius transporto priemonei, gaunamos dvi magnetinės signatūros, tarp kurių galima rasti vėlinimą apskaičiuavus kros-koreliaciją.



22 pav. Vėlavimas tarp dviejų signalų bei kros-koreliacijos funkcija su pažymėtu maksimumo tašku [6]

Šis metodas taip plačiau analizuojamas praktiškai, o gauti rezultatai aptariami skyriuje „Eksperimentinė dalis“.

Anksčiau aptartame straipsnyje [21] rašoma apie galimybę aptikti TP panaudojant du AMR jutiklius, tačiau to paties straipsnio autoriai aptaria keletą metodų, kaip apskaičiuoti TP greitį. Vienas iš metodų pagrįstas laiko sinchronizavimu tarp dviejų magnetinio lauko jutiklių. Žemiau pateiktame paveiksle pavaizduotos dvi magnetinės signalūros, kurias sukuria pro jutiklius pravažiuojanti TP.



23 pav. TP aptikimas pagal magnetines signalūras [21]

Šioje metodikoje greičiui apskaičiuoti naudojamas laiko tarpas, kuris gaunamas signalui kertant slenkstinę ribą. Kai slenkstinė reikšmė kertama iš viršaus į apačią pirmą kartą, paleidžiamas laikmatis. Signalui kirtus slenkstinę reikšmę antrą kartą iš viršaus į apačią, laikmatis sustabdomas. Antroji slenkstinė reikšmė būna šiek tiek aukščiau nei pirmoji. Jos parinkimas nustatomas empiriškai. Tai atliekama su abejomis signalūromis. Tokiu būdu gaunami du laikai Δt_{UP} ir Δt_{DOWN} . Turint šiuos laikus, nesunkiai galima apskaičiuoti TP greitį:

$$\Delta t_{UP} = t_{B,UP} - t_{A,UP} \quad (4)$$

$$\Delta t_{DOWN} = t_{B,DOWN} - t_{A,DOWN} \quad (5)$$

$$v_{UP} = \frac{S_{AB}}{\Delta t_{UP}} \quad (6)$$

$$v_{DOWN} = \frac{S_{AB}}{\Delta t_{DOWN}} \quad (7)$$

$$v = \frac{v_{UP} + v_{DOWN}}{2} \quad (8)$$

Šiose formulēsē S_{AB} yra atstumas tarp magnetinio lauko jutiklių A ir B. Kadangi jokie jutikliai nebūna tobulai vienodi, atsiranda tam tikro dydžio paklaida dėl sensorių nevienodumo. Ši nevienodumą galima pašalinti pritaikius tam tikrą konstantą ε . Šio faktoriaus įtaka parodoma žemiau pateiktomis formulėmis:

$$\overline{t_{UP}} = \Delta t_{UP} - \varepsilon \quad (9)$$

$$\overline{t_{DOWN}} = \Delta t_{DOWN} + \varepsilon \quad (10)$$

$$\overline{v_{UP}} + \overline{v_{DOWN}} = \frac{S_{AB}}{\Delta t_{UP} - \varepsilon} + \frac{S_{AB}}{\Delta t_{DOWN} + \varepsilon} = \frac{S_{AB}(\Delta t_{UP} - \varepsilon + \Delta t_{DOWN} + \varepsilon)}{\Delta t_{UP}\Delta t_{DOWN} + \varepsilon(\Delta t_{UP} - \Delta t_{DOWN}) + \varepsilon^2} \quad (11)$$

Jeigu $\varepsilon \rightarrow 0$ gauname:

$$\overline{v_{UP}} + \overline{v_{DOWN}} = \frac{S_{AB}(\Delta t_{UP} + \Delta t_{DOWN})}{\Delta t_{UP}\Delta t_{DOWN}} \quad (12)$$

Iš čia matome, kad nuo faktoriaus ε priklauso apskaičiuoto greičio tikslumas. Taip pat matome, kad laiko sinchronizavimas (ε mažinimas) yra vienas iš būdų pagerinti greičio apskaičiavimo tikslumą.

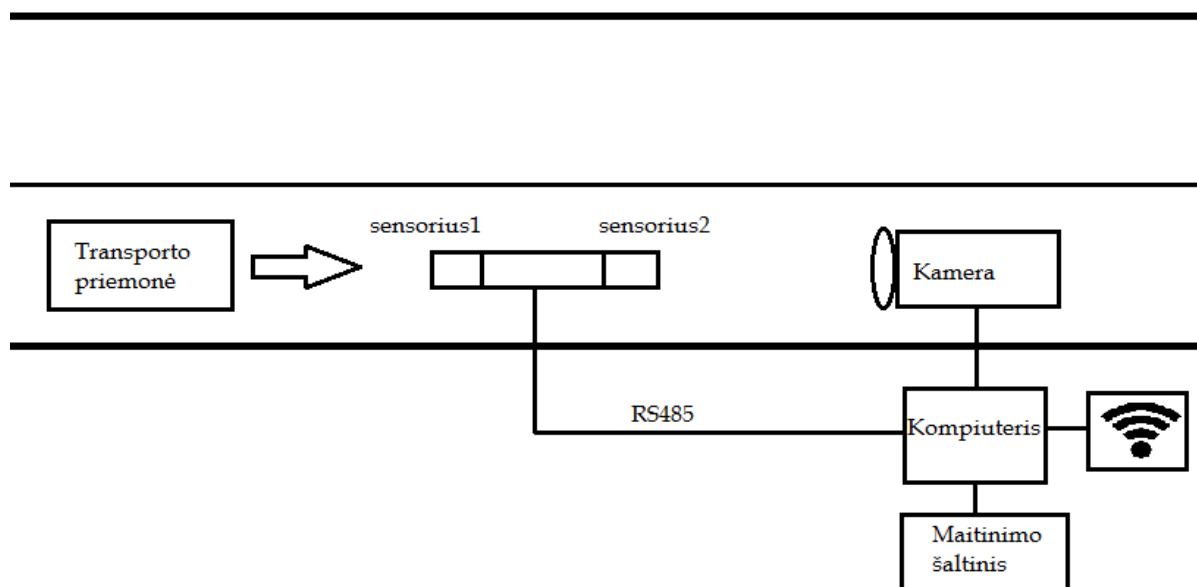
Apibendrinant analizuotą literatūrą galima teigti, kad magnetinio lauko jutiklių naudojimas transporto priemonių aptikimui ir greičio nustatymui bei klasifikavimui yra aktuali tema pasaulyje. Šia tema vykdomi įvairūs eksperimentai, kuriamos bevielės sistemos, kurios sugebėtų ilgą laiką veikti iš baterijų ir stebėti transporto priemonių srautas. Pagrindinė problema, su kuria susiduriama visuose analizuotuose literatūros šaltiniuose yra tikslus transporto priemonės greičio nustatymas panaudojant magnetinio lauko jutiklių duomenis. Pagrindinis metodas, kuriuo bandoma nustatyti transporto priemonės greitį yra kros-koreliacija tarp dviejų signalų, gautų iš dviejų magnetinių sensorių, stovinčių tam tikru atstumu vienas nuo kito. Tačiau norint turėti tokiu principu efektyviai veikiančias sistemas reikia apskaičiuoti transporto priemonės greitį kaip įmanoma greičiau, o kros-koreliacija yra

lėtas metodas tam atlikti. Akivaizdu, kad yra poreikis išsiaiškinti, kaip galima paspartinti užfiksuotos transporto priemonės greičio apskaičiavimą ir tuo pačiu turėti gerą matavimo tikslumą. Būtent todėl sekančiame skyriuje atliekama detali vėlinimo tarp dviejų signalų nustatymo būdų analizė, kurios rezultatų pagrindu būtų sukurtas naujas, didelio tikslumo ir greitai veikiantis algoritmas, kurio dėka būtų apskaičiuotas transporto priemonės greitis.

2. VĒLINIMO TARP DVIEJŲ SIGNALŲ NUSTATYMO BŪDAI

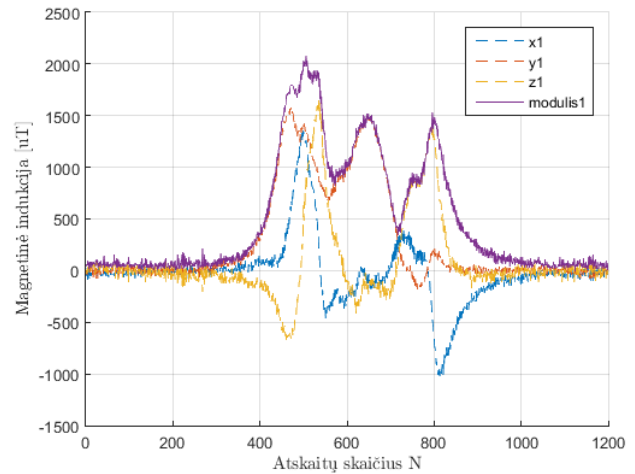
Transporto priemonių greičio nustatymas yra viena iš daugelio transporto srautų valdymo sudedamųjų dalių. Šioje dalyje aptariami būdai, kurie buvo naudojami norint surasti vėlinimą tarp dviejų signalų. Tai reikalinga tam, kad galiausiai galėtume sužinoti transporto priemonės greitį.

Duomenų surinkimui buvo sukurta ir realioje aplinkoje įdiegta sistema, kuri užregistruoja transporto priemonių magnetinio lauko iškraipymą. Sistemos vaizdas pateiktas 24 pav. Sistemą sudaro keturi magnetinio lauko jutikliai (dvi poros po du jutiklius), vienas nuo kito išdėstyti 30 cm atstumu. Juos valdo mikrovaldiklis. Iš viso naudojami 4 magnetiniai jutikliai tam, kad būtų pasiektas 2 kHz diskretizavimo dažnis, nes vienas magnetinis jutiklis gali atiduoti duomenis 1 kHz dažniu. Pastačius antrą jutiklį ir apklausinėjant jutiklius vieną paskui kitą galime gauti 2 kHz duomenų surinkimo dažnį. Duomenys RS485 sąsaja yra perduodami į kompiuterį. Kompiuteris turi interneto ryšį ir prie jo galima bet kada prisijungti. Taip pat kompiuteris turi grafinę vartotojo sąsają, kurią prisijungus galima stebėti daugybę sistemos parametrų ir realiu laiku stebėti sistemos veikimą. Be to, kompiuteris valdo kamerą, kuri fotografuoja pravažiuojančias transporto priemones. Transporto priemonių fotografavimas vykdomas pagal magnetinio lauko jutiklių atsiųstus duomenis.



24 pav. Sistemos, naudojamos duomenų surinkimui ir testavimui struktūra

Sistemos dėka galima gauti ne tik bet kokios transporto priemonės magnetines signatūras, bet ir nufotografuoti transporto priemonę. Taip žinome, kokia transporto priemonė sukėlė atitinkamus magnetinio lauko iškraipymus. Žemiau esančiame paveiksle pavaizduota transporto priemonė kurią nufotografavo sistema ir šalia pateikiamas vaizdas, kaip buvo iškraipytos magnetinio lauko dedamosios x, y, z . Taip pat pateikiamas apskaičiuotas šių ašių modulis.



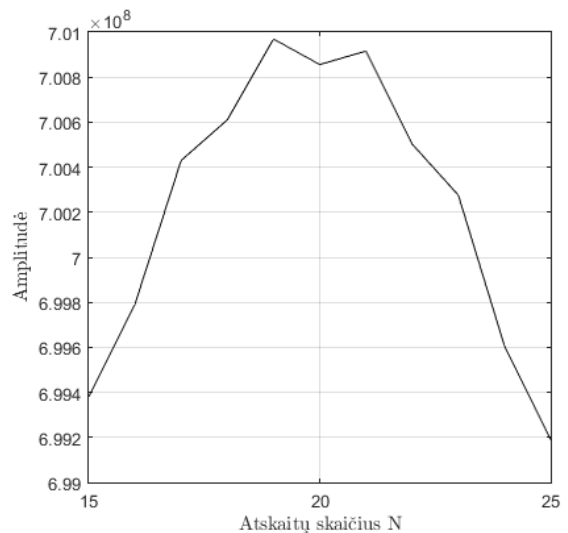
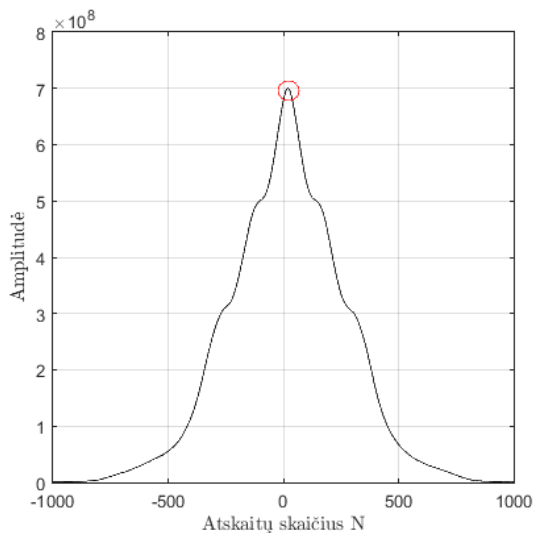
25 pav. Transporto priemonė ir jos sukeltas magnetinio lauko iškreipymas

2.1 Kros-Koreliacija

Kros-koreliacija yra pirmasis metodas, kuris analizuojamas šiame darbe. Tai yra vienas seniausių, žinomiausių ir labiausiai paplitusių metodų, kurie yra naudojami kuomet reikia surasti vėlavimą tarp dviejų signalų. Kros-koreliacija tarp dviejų signalų apskaičiuojama pagal žemiau pateiktą formulę.

$$CC(l) = \sum_{n=1}^N S_1(n) \cdot S_2(n-l), \quad \text{kurl} = -N, -(N-1), \dots, 0, \dots, (N-1), \quad (13)$$

Formulėje (12) N yra signalą sudarančių atskaitų skaičius. Iš formulės matyti, kad kros-koreliacija yra ne kas kita kaip daugyba tarp kiekvienos signalą sudarančios atskaitos, signalą vis perstumiant per vieną poziciją į vieną arba kitą pusę. Pritaikius šią formulę dviems signalams, gaunama kros-koreliacinė funkcija. Ši funkcija pavaizduota 26 pav. Ji gauta naudojant realius signalus, kurie pavaizduoti 25 pav.



26 pav. Kros-koreliacinė funkcija. Kairėje - pilnas jos vaizdas, dešinėje - išdidintas vaizdas ties maksimumo vieta

Turint kros-koreliacinę funkciją, nesunku apskaičiuoti vėlinimą tarp signalų. Jis apskaičiuojamas pagal žemiau pateiktą formulę (14).

$$\Delta l = \arg \max (CC(l)) \quad (14)$$

Iš aukščiau pateiktos formulės matome, kad vienintelis dalykas kurį reikia atlikti, norint surasti vėlinimą tarp signalų iš kros-koreliacinės funkcijos yra surasti šios funkcijos maksimalios reikšmės indeksą. Rastas maksimalios reikšmės indeksas atitinka suvėlinimą tarp signalų. Tuomet apskaičiuoti transporto priemonės greitį galima pagal žemiau pateikiamą formulę:

$$v = \frac{s}{t} = \frac{s}{\Delta l \cdot t_s}; \quad (15)$$

čia formulėje s yra nuvažiuotas kelias, arba mūsų atveju atstumas tarp jutiklių, Δl yra vėlinimas atskaitomis tarp dviejų signalų, o t_s yra diskretizavimo periodas. Pavyzdžiui, aukščiau pateiktame pavyzdyje gautas $\Delta l = 19$. Atstumas tarp jutiklių $s = 0.3m$, o diskretizavimo dažnis $f_s = 2000Hz$,

vadinasi diskretizavimo periodas $t_s = \frac{1}{f_s} = \frac{1}{2000Hz} = 5 \cdot 10^{-4} s$. Iš čia seka, kad duotame pavyzdyje

nagrinėjamos transporto priemonės greitis apytiksliai yra apskaičiuojamas pagal (15) formulę:

$$v = \frac{0.3m}{19 \cdot 5 \cdot 10^{-4} s} \approx 31,6 \frac{m}{s}$$

2.2 Masės centro radimas

Masės centro radimas pagrįstas seniai žinomu metodu, kuriuo apskaičiuojamas uždara kreivė ribojamos figūros plotas. Norint pritaikyti šį metodą praktikoje, reikalinga tam tikra ribinė vertė, kurią viršijus užfiksuojama figūros pradžia, nes jeigu skaičiuotume masės centrą su signalu, kuris turi „uodegėles“, gautume netikslumus. Norint, kad šis metodas veiktų efektyviai, reikalingas tam tikras ribinės vertės nustatymas, kad gautume mažiausią triukšmų įtaką. Jeigu turime du diskretinius signalus $S_1(n)$ ir $S_2(n)$, tai kiekvienos figūros, kuri gaunama iš šių diskretinių signalų masės centrą galima surasti pagal sekančia formulę:

$$n_1 = \frac{\sum_{n=1}^N S_1(n) \cdot n}{\sum_{n=1}^N S_1(n)}, \quad n_2 = \frac{\sum_{n=1}^N S_2(n) \cdot n}{\sum_{n=1}^N S_2(n)}; \quad (16)$$

čia n_1 ir n_2 yra indeksai, kurie žymi apskaičiuotos figūros masės centrą, o N yra signalą sudarančių atskaitų skaičius, kuris priklauso nuo panaudojamos ribinės vertės. Kaip jau minėta aukščiau, šis metodas yra labai jautrus pasirenkamai ribinei vertei, nuo kurios yra fiksuojama signalo pradžia ir užfiksuojamos uždara kreivė apribotos figūros. Eksperimentinėje dalyje bus plačiau aptarta ribinės reikšmės pasirinkimo įtaka gaunamam rezultatui. Iš aukščiau pateiktų masės centrų formulių žinome, kad tai yra masės centrų indeksai. Tuomet juos turėdami galime pagal formulę (17) apskaičiuoti transporto priemonės greitį:

$$v = \frac{s}{t} = \frac{s}{(n_2 - n_1) \cdot t_s}; \quad (17)$$

čia s – nuvažiuotas kelias, arba mūsų atveju atstumas tarp jutiklių, $n_2 - n_1$ yra vėlinimas atskaitomis tarp dviejų signalų, t_s – yra diskretizavimo periodas. 27 pav. galime matyti, kokia signalų dalis naudojama masės centro skaičiavimui. Vienam signalui skaičiavimo intervalą užduoda ribos, pavaizduotos vertikaliomis tiesėmis b1-b2, kitam signalui – c1-c2.

Signalui viršijus ribinę vertę kylant ir krentant, gaunama uždara figūra, kurią riboja signalo forma ir pagrindas, kurio vertė yra lygi ribinei vertei. Tuomet pagal (16) formulę apskaičiuojamas masės centras ir gaunama vienos signalatūros masės centro reikšmė. Kitai signalatūrai naudojama ta pati ribinė vertė, tačiau akivaizdu, kad tos signalatūros indeksai ir reikšmės bus kitokie dėl to, kad antroji signalatūra atsilieka nuo pirmosios, todėl tie indeksai gausis kitokie. Apskaičiavus antrosios signalatūros masės

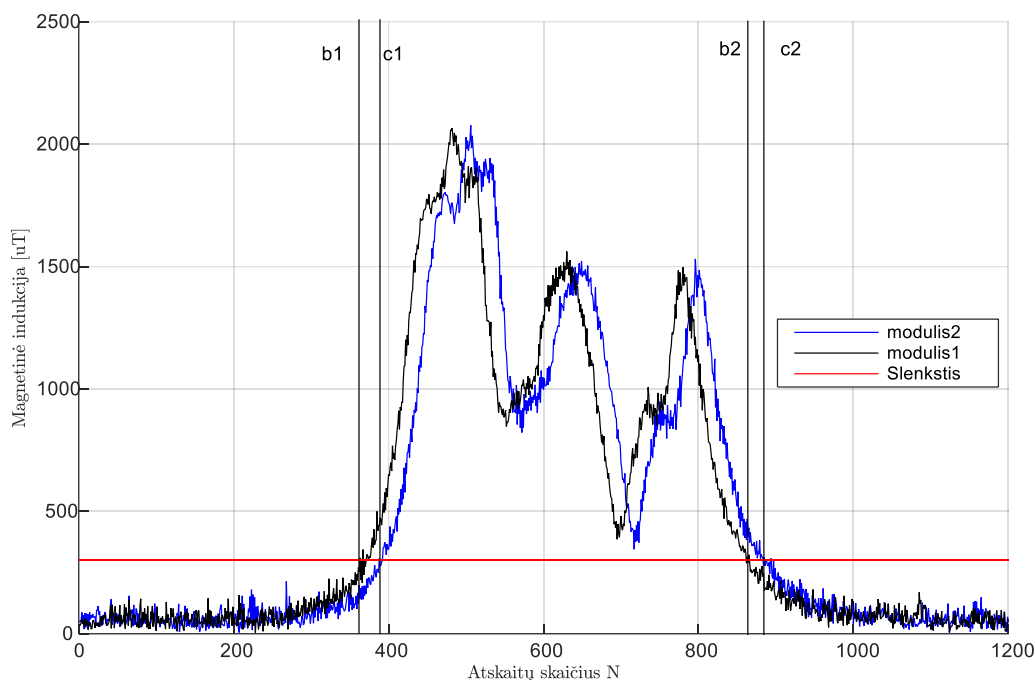
centrą, galima surasti skirtumą tarp masių centrų ir pagal (17) formulę apytiksliai apskaičiuoti transporto priemonės greitį.

2.3 Figūros, apribotos slenkstinėmis vertėmis, centro skaičiavimas

Šio metodo esmė yra labai panaši į masių centrų radimą, tik šiuo atveju nereikia skaičiuoti jokių sumų, o užtenka užfiksuoti indeksus, kuriuose yra kertamos ribinės vertės. Kaip ir ankstesniais atvejais, reikia surasti ribinėmis vertėmis apribotos figūros centrą. Taigi galime pritaikyti žemiau pateiktą formulę ir pagal ją surasti figūros centrą. Šio metodo esmę ir formulę paaiškinantis paveikslukas pateiktas 27 pav.

$$n_1 = \frac{b_2 - b_1}{2}, \quad n_2 = \frac{c_2 - c_1}{2} \quad (18)$$

Pagal formulę (18) suradę kiekvienos figūros centrą galime nesunkiai apskaičiuoti vėlinimą tarp signalų, kuris bus lygūs tiesiog skirtumui, tarp surastų figūrų centrų indeksų. Palyginti su masių centrų metodu, šis metodas yra žymiai greitesnis ir jam nereikia atlikti jokių skaičiavimų, užtenka surasti vertes, kur kertama ribinė vertė, kurią galima užsiduoti.



27 pav. Figūros, apribotos slenkstinėmis vertėmis, kurios reikalingos figūros centrui apskaičiuoti

Iš aukščiau pateiktos figūros matome, kad vos tik viena ir kita signalūra kerta užsiduotą ribinę reikšmę, yra užfiksuojamas indeksas ar atskaitos numeris, ties kuriuo tai įvyko. Abejoms signalūroms

šis indeksas bus skirtingas. Baigiantis signalui jo pabaigoje riba yra kertama iš viršaus į apačią ir vėl užfiksuojami signalo indeksai, ties kuriais ši ribinė reikšmė kertama.

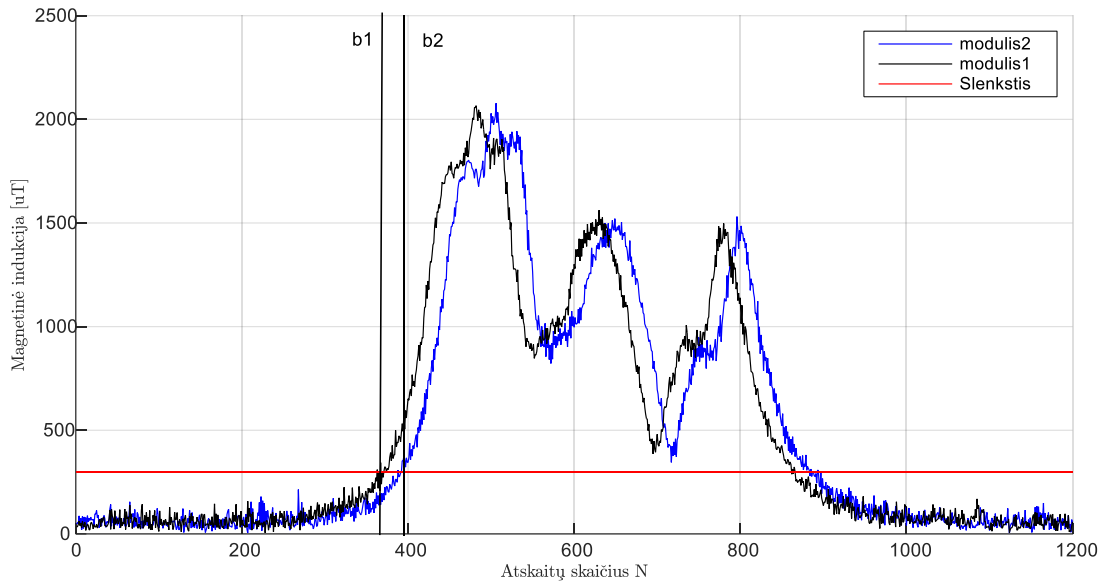
Palyginus šį metodą su masių centrų ar kros-koreliacijos metodu, galima išvelgti pagrindinius šio metodo privalumus prieš anksčiau aptartus metodus: nereikia saugoti viso signalo mikrovaldiklio atmintyje, užtenka sekti, kuomet yra kertamos užsiduotos ribinės vertės. Kros-koreliacijos ir masės centro metodų atvejais reikalinga atlikti labai daug skaičiavimų mikrovaldiklyje. Tie algoritmai skaičiavimo prasme yra sudėtingi ir užima nemažai laiko. Žinoma, kros-koreliacijos metodas laikine prasme užtrunka ilgiausiai ir yra pats lėčiausias, tačiau duoda geriausią rezultatą. Masės centro metodas yra žymiai greitesnis už kros-koreliaciją, tačiau lėtesnis už figūros centro, apribotos ribinėmis vertėmis metodą.

Nors šis metodas ir yra labai paprastas, eksperimentinėje dalyje plačiau nagrinėjami šio metodo trūkumai, nes šis metodas turi vos keletą privalumų lyginant su kitais metodais, t.y, nereikia mikrovaldiklio atmintyje saugoti signalo įrašo ir signalui pasibaigus iškart galime turėti atsakymą, koks yra vėlavimas tarp signalūrų, antru atveju mums nereikia atlikinėti daugybės ir sudėties operacijų, t.y. nereikalingi absoliučiai jokie matematiniai skaičiavimai. Vienintelė naudojama matematinė operacija yra vienas reikšmės palyginimas su kita reikšme, kas yra labai greita ir efektyvu, todėl šis metodas leidžia tarsi realiu laiku žinoti atsakymą, koks yra vėlavimas tarp signalūrų. Tačiau šis metodais yra labai priklausomas nuo pasirinkto sistemos diskretizavimo dažnio (mažinant diskretizavimo dažnį prarandama sistemos rezoliucija), taip pat keičiant slenkančio vidurkio parametrus keičiasi galutinis atsakymas, nes pasikeičia indeksų reikšmės, kuriuos signalų signalūros kerta užsiduotas ribines reikšmes. Plačiau apie vėlinimo tarp dviejų signalų nustatymo būdo trūkumus ir privalumus bei atliktus eksperimentus bus pateikta informacijos eksperimentinėje dalyje.

2.4 Slenkstinės reikšmės viršijimo metodas

Šis metodas yra labai panašus į jau aprašytą skyrelyje 2.10. Vienintelis skirtumas tarp šių metodų yra tas, kad aukščiau aptartu metodu ieškoma, kuomet signalūros kerta ribines vertes tiek signalo priekyje, tiek signalo pabaigoje. Šiuo atveju ieškoma indekso, ties kuriuo kertama slenkstinė reikšmė signalo pradžioje. Tuomet vėlinimas tarp signalų atskaitomis gali būti apskaičiuojamas pagal žemiau esančią formulę:

$$\Delta l = b_2 - b_1. \quad (19)$$



28 pav. Paveikslas paaškinantis slenksstinės vertės viršijimo metodą

Šis metodas turi tokius pačius privalumus prieš kros-koreliacijos ir masės centro metodą kaip ir aukščiau aptartas figūros, apribotos ribinėmis vertėmis centro skaičiavimo būdas. Lyginant su pastaruoju metodu, šio metodo skaičiavimo algoritmas dar lengvesnis. Reikia paprasčiausiai aptikti, kuomet viena ir kita signalūra viršija užduotą ribinę vertę ir užfiksuoti tuos indeksus. Skirtumas tarp šių indeksų ir yra vėlavimas tarp signalų atskaitomis. Tuomet transporto priemonės greitį galime apskaičiuoti pagal aukščiau aprašytą (19) formulę. Šio metodo privalumas yra tas, kad kaip ir pastaruoju atveju mikrovaldiklyje nereikia išsaugoti viso signalo atskaitų, tereikia žinoti ribinę vertę, kurią valdome patys.

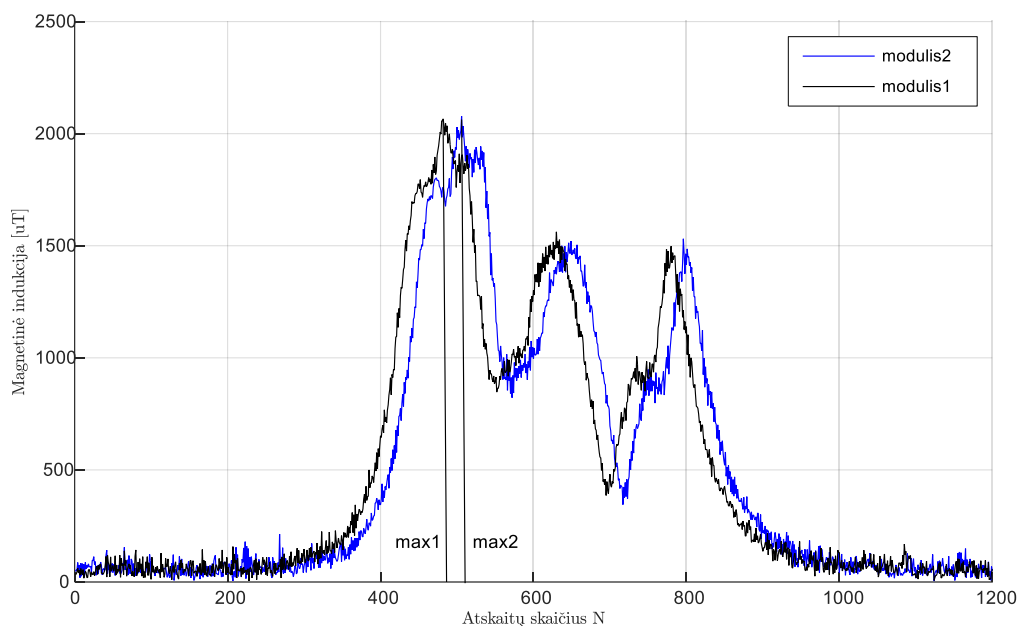
Kaip ir kiti metodai, šis metodas yra labai jautrus slenkančio vidurkio parametrui bei diskretizavimo dažniui. Abu šie veiksniai įtakoja metodo tikslumą ir reikia į juos atsižvelgti parenkant optimalią ribinę vertę jeigu naudojamas šis metodas. Tačiau visumoje šis metodas nėra toks tikslus kaip masės centro ar kros-koreliacijos metodas.

2.5 Atstumo tarp maksimumų skaičiavimas

Naudojant šį metodą reikia išsaugoti visą signalą į mikrovaldiklio atmintį, arba sekti kaip keičiasi signalo reikšmės ir pagal tam tikrus kriterijus fiksuoti, kuomet galėjo būti pasiektas maksimumas ir tada surasti to maksimumo indeksą. Taip maksimumų indeksai randami abejoms signalūroms ir tuomet tų maksimumų skirtumas yra vėlavimas tarp dviejų signalų. Taigi jeigu turime dvi diskretinių signalų sekas S_2 ir S_1 aprašyti galima pagal žemiau pateikiamą formulę:

$$\Delta l = \arg \max(S_2) - \arg \max(S_1) \quad (20)$$

Žemiau esančiame paveiksle pavaizduota šio metodo esmė su pažymėtomis reikšmėmis.



29 pav. Paaškinamasis paveikslas, kaip naudojamas atstumo tarp maksimumų metodas

Kaip jau minėta aukščiau, šio skaičiavimo būdo trūkumas yra tas, kad reikia saugoti mikrovaldiklio atmintyje visą ar bent jau dalį signalūros. Taip pat šiam metodui reikalingas kuo didesnis sistemos diskretizavimo dažnis, nes kuo šis dažnis mažesnis, tuo lėkštesnės bus signalo viršūnės, o tai savo ruožtu įtakos prastesnį galutinį rezultatą. Užfiksavus šiuos maksimumų indeksus ir radus skirtumą tarp jų, pagal (15) formulę galima apskaičiuoti transporto priemonės greitį.

Šiam metodui taip pat būdinga didelė įtaka naudojant slenkančią vidurkį. Kuomet signalas yra nefiltruojamas ir naudojami tokie duomenys, signale matomas triukšmas. Šis triukšmas labai trukdo norint tiksliai surasti maksimumo indeksus, tačiau signalą nufiltravus per didelės eilės slenkančiu vidurkiu gaunamas kitoks rezultatas ir tampa neaišku, kuris rezultatas yra teisingesnis.

Kaip ir aukščiau aptarti vėlinimo tarp dviejų signalų radimo būdai, šis būdas yra jautrus slenkančio vidurkio ir diskretizavimo dažnio parametrams. Apskritai diskretizavimo dažnio įtaka turi didelę įtaką šių metodų rezultatams dėl to, kad mažėjant diskretizavimo dažniui, mažėja sistemos rezoliucija.

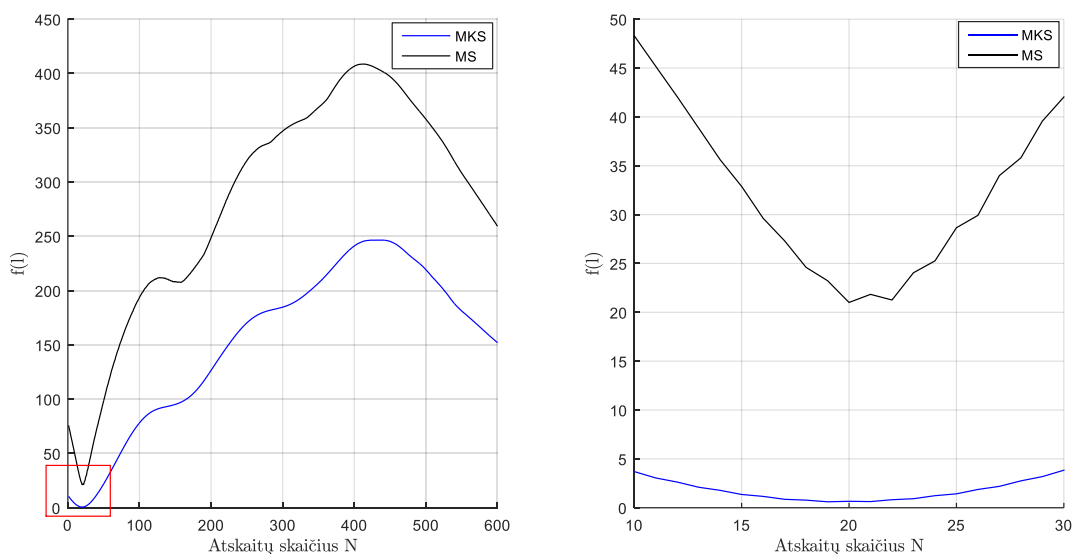
2.6 Mažiausio kvadratų skirtumo metodas ir mažiausio skirtumo skaičiavimas

Tai jau labai senai žinomas ir paplitęs metodas naudojamas statistinės analizės metoduose. Šio metodo esmė, kaip ir kros-koreliacijos metodo atveju, kad turint vieną signalūrą, reikia ją visą prastumti per kitą signalūrą ir kiekviename taške suskaičiuoti skirtumo kvadratą, arba tiesiog skirtumą. Tokiu būdu gaunamas funkcijos grafikas, o tame grafike reikia surasti funkcijos minimumo

indeksą. Funkcijos minimumo indeksas parodo, koks yra vėlavimas tarp dviejų signalų. Žemiau pateikiama formulė, kaip apskaičiuojamas randama mažiausio kvadratų skirtumo funkcija jeigu turime dvi diskretinių signalų sekas S_1 ir S_2 .

$$f(l) = \sum_{n=1}^N (S_1(n) - S_2(n-l))^2, \quad kurl = -N, -(N-1), \dots, 0, \dots, (N-1), \quad (21)$$

Panaudojant pavyzdinių signalų signatūras, galime apskaičiuoti šios pavyzdinės signatūros skirtumo kvadratų funkciją. Jos vaizdą pateikiame žemiau:



30 pav. Mažiausių kvadratų skirtumo ir mažiausio skirtumo skaičiavimo rezultatas grafikuose. Kairėje vieno signalo ilgio signatūros funkcija, o dešinėje pritrauktas funkcijos vaizdas ties jos minimumo vieta

Aukščiau pateiktame paveiksle palyginami abu skaičiavimo būdai: mažiausio kvadratų skirtumo ir mažiausio skirtumo. Abu šie metodai skaičiuojami iš esmės taip pat ir turi tuos pačius privalomus bei trūkumus, vienintelis skirtumas tas, kad kai skirtumas keliamas kvadratu tai gaunamos žymiai mažesnės reikšmės ir funkcija tampa lėkštesne. Tuo tarpu jeigu skirtumas kvadratu nekeliamas tai funkcija gaunama su tam tikrais „aštriais“ kampais. Abiem atvejais gavus funkcijos grafiką, reikia apskaičiuoti minimumo indeksą. Jis yra lygus vėlinimui tarp signalų. Tai atliekame pagal žemiau pateiktą formulę.

$$\Delta l = \arg \min (f(l)) \quad (22)$$

Šio skaičiavimo būdo trūkumas kaip ir kros-koreliacijos ar masių centrų skaičiavimo atveju yra tas, kad mikrovaldiklio atmintyje reikia išsaugoti visą signalo įrašą, taigi naudojama mikrovaldiklio atminties resursai, o algoritmas gali pradėti veikti tik tada kai jau galutinai turima visa signalo

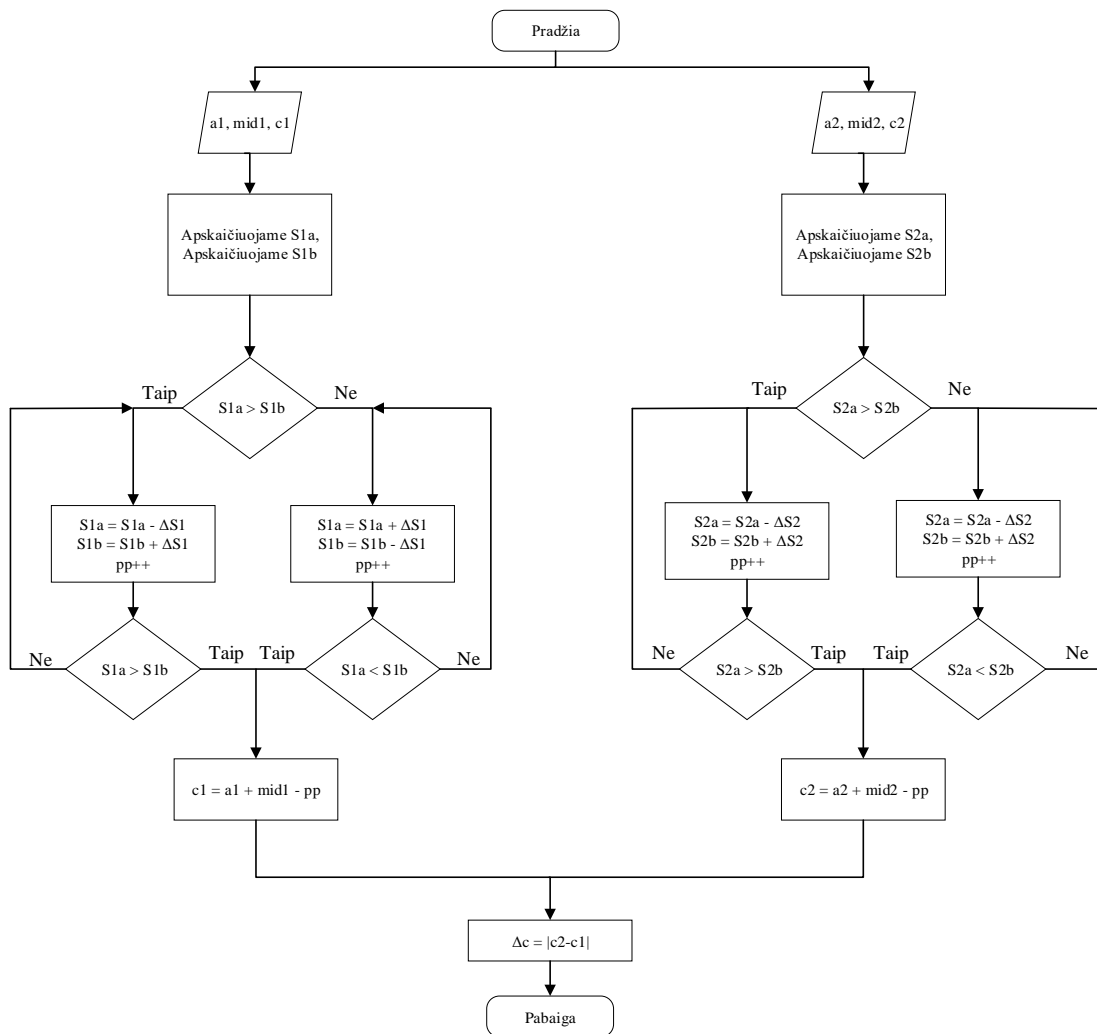
signatūra. Tačiau palyginti su kros-koreliacija ar masių centrų metodu šis metodas skaičiavimo prasme yra gerokai spartesnis ir lengviau realizuojamas mikrovaldiklyje, o surasti funkcijos minimumą nėra sudėtingas uždavinys.

2.7 Vienodų plotų metodas

Šis metodas yra sudėtingas skaičiavimo prasme mikrovaldikliui ir turi trūkumą, kad reikalinga išsaugoti visą signalą į mikrovaldiklio atmintį prieš pradėdant apdoroti signalą šiuo algoritmu. Iš esmės šis metodas yra detalesnis figūros, apribotos ribinėmis vertėmis, centro metodo variantas, mat šiam algoritmui įgyvendinti reikia atlikti visus aukščiau minėtame algoritme pateiktus žingsnius. Iš pradžių reikia surasti abiejų signatūrų indeksus, ties kuriomis signalai kerta ribinę reikšmę ir indeksus, ties kuriomis signatūros kerta ribinę reikšmę iš viršaus į apačią, Tuomet yra apskaičiuojamos gautų figūrų plotas pagal Riemano sumos formulę:

$$S1a = \sum_{n=i1}^{i2} B_1(n) \cdot \Delta n, \quad S2a = \sum_{n=c1}^{c2} B_2(n) \cdot \Delta n, \quad (23)$$

Šioje formulėje $i1$ ir $i2$ bei $c1$ ir $c2$ ir signatūrų indeksai, kuomet kertamos ribinės reikšmės iš apačios ir iš viršaus. Tuomet kaip ir anksčiau minėtu metodu surandami figūros centrai ir figūra yra padalinama į dvi dalis. Suskaičiuojamos abiejų dalių sumos ir toliau bandoma tas sumas išlyginti, arba gauti mažiausią skirtumą tarp jų. Toliau esančiame paveiksle pateikiamas šio metodo algoritmas.



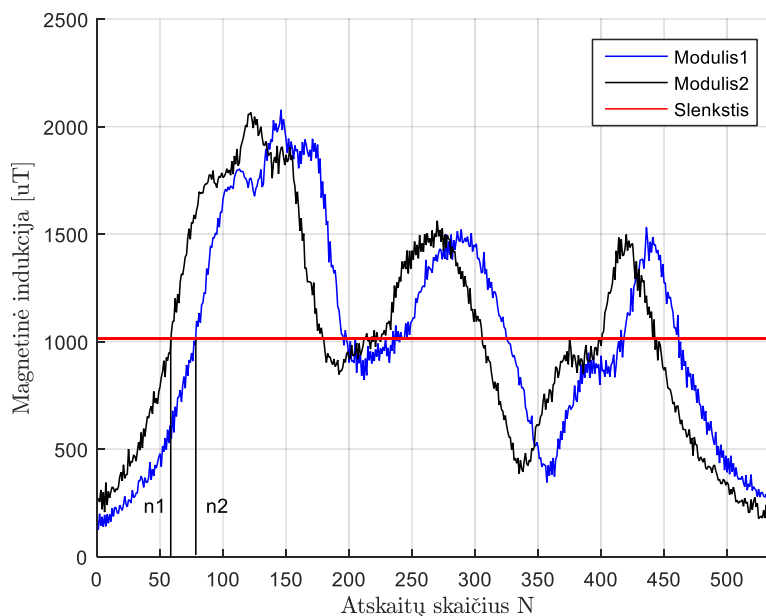
31 pav. Vienodų plotų skaičiavimo metodo algoritmas

Iš pateikto algoritmo matome, kad iš pradžių randamas figūros centro indeksas, o figūra padalinama į dvi dalis ir apskaičiuojami tų figūrų plotai. Toliau atliekamas ciklas tol, kol figūrų plotai išsilygina, arba plotų skirtumas tampa minimalus. Tada paskaičiuojamas naujas centro indeksas, kuris parodo kur turėtų būti centras, kad figūros plotai gautūsi vienodi. Tas pats atliekama su antru signalu. Tuomet skirtumo modulis tarp apskaičiuotų figūrų centrų yra lygus vėlinimui tarp dviejų signalų.

2.8 Slenkstinės vertės viršijimo radimas, kai slenkstis lygus pirmojo signalo vidurkiui

Šis skaičiavimo būdas nelabai kuo skiriasi nuo anksčiau aprašytų slenkstinės vertės viršijimų metodų. Vienu atveju slenkstinė vertė gali būti parenkama empiriškai, kitu atveju vidurkinant sensoriaus išėjimo signalą ir parenkant slenkstinę reikšmę keletu standartinių nuokrypių toliau nei vidurkio reikšmė. Skaičiuojant vėlinimą tarp dviejų signalų šiuo būdu, slenkstinė vertė yra prilyginama pirmojo signalo vidurkio reikšmei. Dėl to, slenkstinė vertė būna ganėtinai didelė ir gaunama signatūra dažniausiai būna žymiai trumpesnė nei originalus signalas. Dėl to sutrumpėja

skaičiavimo laikas ir išsaugoma mikrovaldiklio atmintis, man nereikia saugoti labai daug signalo atskaitų. Reikia pabrėžti, kad šis metodas taikomas jau turint magnetinę signatūrą, t.y. pradžioje vis tiek naudojamas kitoks būdas, kuriuo iš realiu laiku ateinančių duomenų nustatoma slenkstinė vertė, kuomet viršijus pradedamas signalo atskaitų įrašymas. Kai įrašymas baigiasi ir turime pilną signalą, su kuriuo galime dirbti, tuomet pritaikome šį metodą.

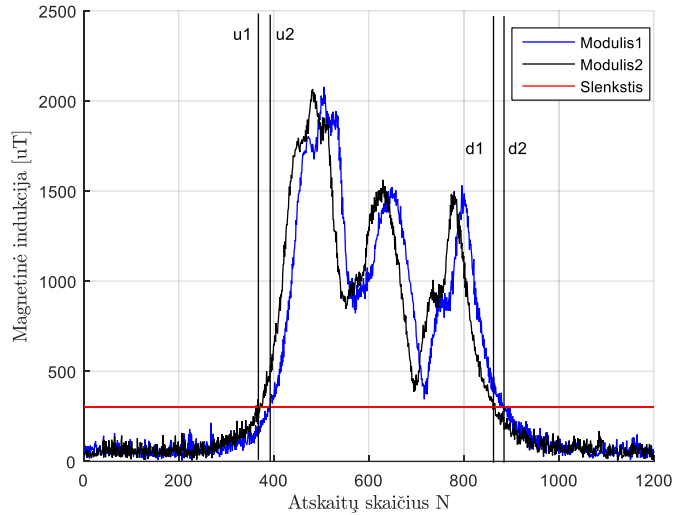


32 pav. Slenkstinės vertės viršijimo, kai slenkstis lygus pirmojo signalo vidurkiui, grafikas

Iš anksčiau pateiktos iliustracijos matyti, kad šiuo atveju vėlinimas tarp dviejų signalų apskaičiuojamas tiesiog $n2-n1$. Taip pat reikia pabrėžti, kad čia dirbame su nefiltruotu signalu. Pritaikius slenkamojo vidurkio filtrą rezultatai greičiausiai būtų dar geresni, o interpoliavus signalą zonoje, kurioje kertama slenkstinė reikšmė, gautume geresnę rezoliuciją nei 1 atskaita.

2.9 Figūros, apribotos slenkstinėmis vertėmis iš abiejų pusių, vidurkio skaičiavimas

Tai dar vienas skaičiavimo būdas, kuriam reikalinga užduoti slenkstinę reikšmę. Kai kylantis signalo frontas kerta slenkstinę reikšmę, išsaugomi indeksai, ties kuriais tai įvyko. Tuomet laukiama, kol signalas iš viršaus kirs slenkstinę vertę keliaujant žemyn. Tuomet apskaičiuojamas skirtumas atskaitomis, kuomet kylantis signalo frontas kirto slenkstinę reikšmę ir skirtumas atskaitomis, kuomet krintantis signalo frontas kirto slenkstinę reikšmę. Iš šių dviejų skirtumų išvedamas vidurkis, kuris ir yra galutinis rezultatas.



Pav. 32. Pažymėti taškai, kurie naudojami aukščiau aptarto metodo skaičiavimui.

Vėlinimas tarp dviejų signalų šiuo atveju skaičiuojamas sekančiais:

$$\Delta n = \frac{(u_2 - u_1) + (d_2 - d_1)}{2} \quad (24)$$

Šio skaičiavimo būdo didžiausias privalumas palyginti su kitais metodais yra tas, kad nereikia išsaugoti visų signalų mikrovaldiklio atmintyje. Šiuo metodu apskaičiuoti vėlinimą tarp signalų galima realiu laiku, kai transporto priemonė važiuoja virš magnetinio lauko jutiklio. Vos tik ji pravažiuoja, mes jau turime atsakymą ir nebereikalingas joks tolimesnis signalų apdorojimas. Taip ženkliai sutaupome mikrovaldiklio resursų. Pagrindinis šio metodo trūkumas tas, kad signalai turėtų būti filtruoti, nes su triukšmingais signalais šis metodas neduoda labai gerų rezultatų, be to, slenkstinė reikšmė turėtų būti gan didelė, nes priešingu atveju galime papulti į triukšmo zoną, kai slenkstinė vertė viršijama daug kartų iš apačios į viršų ir iš viršaus į apačią, ko pasėkoje taip pat gaunami netikslūs rezultatai.

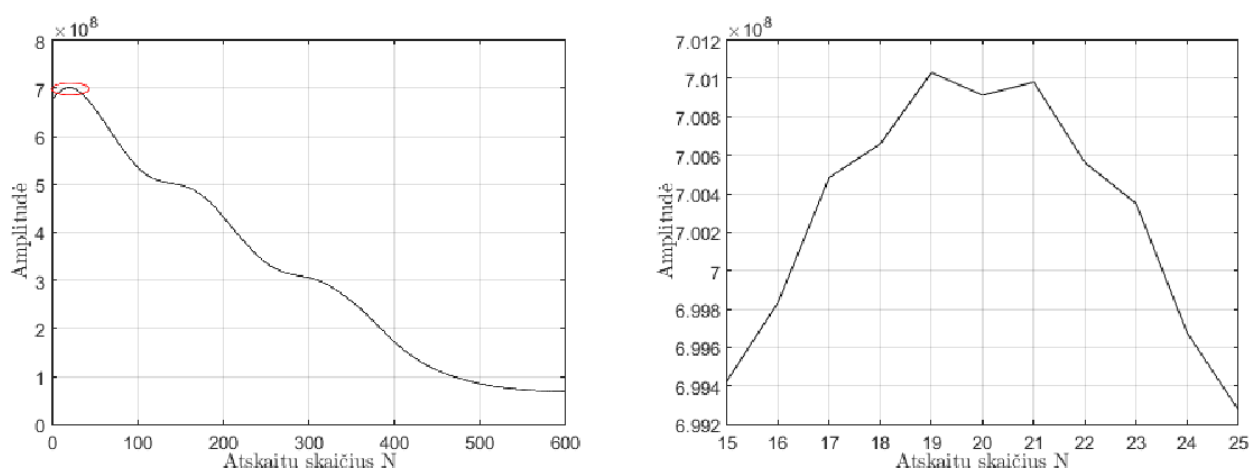
2.10 Figūros, apribotos slenkstinėmis vertėmis, centrų skirtumo skaičiavimas

Šis metodas yra praktiškai toks pats kaip ir aptartas aukščiau, o jam iliustruoti taip pat tinka aukščiau 35 pav. pateiktas paveikslas. Vienintelis skirtumas, kad šiuo atveju apskaičiuojami figūros centrai ir skaičiuojamas tų figūrų centrų skirtumas. Tuomet šiuo atveju vėlinimą tarp dviejų signalų (pagal 35 pav.) galime apskaičiuoti pagal sekančią formulę:

$$\Delta n = \frac{(d_2 - u_1) - (d_1 - u_2)}{2}. \quad (24)$$

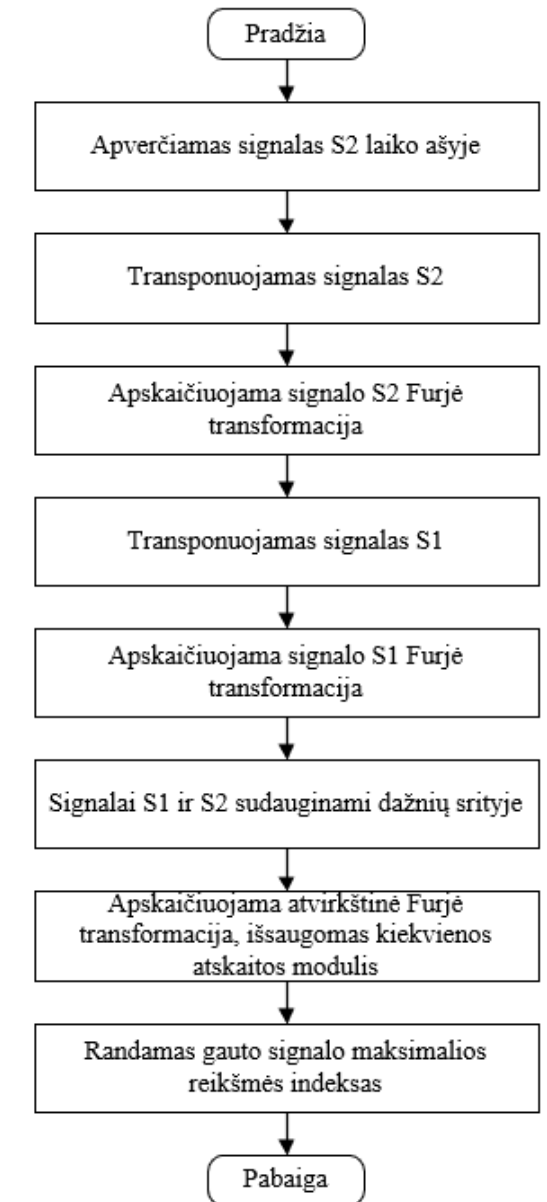
2.11 Suderinto filtro dažnių srityje metodas

Suderinto filtro dažnių ašyje metodas remiasi Furjė transformacijos skaičiavimu (signalas iš laikinės srities perkeliamas į dažnių sritį). Tuomet atliekamas signalo apdorojimas ir signalas gražinamas į laikinę ašį atvirkštinės Furjė transformacijos pagalba. Gautame signale reikia surasti maksimumo reikšmę intervale nuo nulinės atskaitos iki užsiduotos atskaitos, kuri šiuo atveju prilyginama pusei signalo ilgio (tai daroma tam, kad paspartinti skaičiavimus ir neieškoti minimumo visame signalo ilgyje. Metodas yra labai greitas ir paprastai atliekamas.



33 pav. Signalas laikinėje srityje po algoritmo įgyvendinimo (kairėje - kai signalo riba yra pusė signalo ilgio, dešinėje – išdidintas signalo viršūnės vaizdas)

Anksčiau pateiktame paveiksle aiškiai matomas gautas signalo maksimumas. Šį maksimumą turi bet kokios transporto priemonės sukurta magnetinė signatūra ir jis visada būna vienintelis. Kaip matyti, galime dar paspartinti algoritmo skaičiavimo laiką apribodami signalo atskaitų skaičių, kuriame reikia ieškoti maksimumo. Toliau pateikiamas šio skaičiavimo metodo skaičiavimo seka. Skaitoma, kad signalai S1 ir S2 jau išsaugoti ir paruošti tolimesniam apdorojimui.



34 pav. Suderinto filtro metodo apskaičiavimo seka

2.12 Dalinės kros-koreliacijos skaičiavimas

Šis metodas nuo kros-koreliacijos skiriasi tik tuo, kad kros-koreliacija skaičiuojama tik nedidelėje signalo dalyje. Šio metodo, kaip ir kros-koreliacijos metodo, esmė yra surasti kros-koreliacinės funkcijos maksimumo indeksą. Kros-koreliacijos funkcija turi tiek pat taškų, kiek abudu signalai sudėjus. Akivaizdu, kad yra visiškai neefektyvu skaičiuoti kros-koreliacija tiek daug taškų, kai reikia surasti tik tos funkcijos maksimumą. Čia galima panaudoti kitus skaičiavimo būdus, kuriais nustatome vėlinimą tarp signalų. Pavyzdžiui, greitai suskaičiuojamas masės centras – juo surandamas vėlinimas tarp signalų, o tada užsiduodamas tam tikras leistinas nuokrypis į abi puses, ir ties šiomis vėlinimo atskaitomis skaičiuojama kros-koreliacijos funkcijos reikšmė. Suskaičiavus šias reikšmes

ieškomas funkcijos maksimumo indeksas. Žinodami maksimumo indeksą, galime apskaičiuoti vėlinimą tarp signalų. Šio metodo vizualizacija nepateikiama, nes šis metodas yra dviejų metodų sintezė. Kaip pagalbinį metodą, kuris apskaičiuoja pirminį vėlinimą tarp signalų galima naudoti bet kokį metodą, kuris padeda greitai apskaičiuoti vėlinimą tarp signalų, pavyzdžiui slenkstinės vertės viršijimo metodą, arba atstumo tarp maksimumų metodą. Žemiau esančioje lentelėje pateikiamas metodo įvykdymo laikas, esant skirtingiems pagalbiniais metodams.

1 lentelė. Dalinės kros-koreliacijos skaičiavimo būdai:

Pagalbinio vėlinimo radimo būdo pavadinimas	Vėlinimas N (atskaitų skaičius)
Masės centras	19
Atstumas tarp maksimumų	21
Slenkstinės vertės viršijimas	21
Slenkstinės vertės viršimas (slenkstis lygus pirmo signalo vidurkiui)	20
Figūros, apribotos slenkstinėmis vertėmis iš abiejų pusių vidurkio skaičiavimas	21
Figūros, apribotos slenkstinėmis vertėmis, centro skaičiavimas	21
Vienodų plotų skaičiavimas	21

Kaip matyti iš anksčiau pateiktos lentelės, kaip pagalbinį metodą naudojant masių centrą, gaunamas toks pats rezultatas kaip ir taikant kros-koreliacijos skaičiavimo metodą. Vadinas tokiu būdu galima ženkliai paspartinti skaičiavimus. Įdomu tai, kad naudojant visus kitus metodus gautas atsakymas yra lygus 21, tačiau jis nesutampa su originaliu kros-koreliacijos rezultatu. Dėl ko gaunamas toks rezultatas galima matyti iš 2.1 skyrelyje pateikto kros-koreliacijos grafiko. Kros-koreliacijos funkcijos viršūnė yra kampuota, o tokia ji yra dėl to, kad signalams gauti naudojami dviejų jutiklių duomenys, kurie nėra identiški. Iš čia matome, kad šis skaičiavimo metodas yra ne visai tinkamas. Šį metodą reikėtų patobulinti taip, kad jis ieškotų ne pirmojo maksimumo, o turėtų dar šiokią tokį rezervą ir tikrintų toliau esančius taškus, nes dabar algoritmas suranda pirmąjį maksimumą ir tai yra užfiksuojama kaip teisingas rezultatas, tačiau taip apskaičiuotas rezultatas gali būti neteisingas. Antrasis būdas, kaip galima šią problemą išspręsti yra signalų vidurkinimas. Suvidurkinus signalus su tam tikru slenkančiu vidurkiu turėtume gauti tolygias viršūnes ir rezultatai

turėtų sutapti. Plačiau apie slenkančio vidurkio įtaką bus kalbama toliau esančiame 2.14 skyrelyje. Be abejo visi šie metodai žymiai pagreitina skaičiavimus ir toks skaičiavimo metodas yra daug efektyvesnis, nei skaičiuojant kros-koreliaciją per visą signalo ilgį. Tačiau reikia pabrėžti, kad net ir skaičiuojant šiuo atveju mums vis tiek reikia išsaugoti pilnus signalus į mikrovaldiklio atmintį. Nesvarbu, jog skaičiavimai vyksta tik aplink kros-koreliacinės funkcijos viršūnę, tačiau bet kuriame taške rezultatui suskaičiuoti reikalinga turėti pilną signalą. Taigi šiuo metodu galime paspartinti skaičiavimas, tačiau mikrovaldiklio resursų šiuo metodu sutaupyti nepavyks.

2.13 Vėlinimo radimo būdų tarp dviejų signalų apibendrinimas

Žemiau esančioje lentelėje pateikiami visų testuotų metodų rezultatai, kurie aprašyti aukščiau esančiame skyrelyje. Lentelėje pateikiamas metodo pavadinimas, vėlinimas tarp signalų (atskaitų skaičius) ir trukmė, kiek laiko „Matlab“ programiniu paketu užtruko įvykdyti aprašytą metodą. Matavimų trukmė yra dešimties matavimų vidurkis („Matlab“ paketas kiekvieną kartą skirtingai suskaičiuoja trukmę, kurios prireikia algoritmui įgyvendinti).

2 lentelė. Metodų testavimo rezultatai su pavyzdine signalūra

Metodo pavadinimas	Vėlinimas N (atskaitų skaičius)	Skaičiavimo trukmė t, ms
Kros-koreliacija	19	77,4
Masės centro metodas	16,6	117,8
Figūros, apribotos slenkstinėmis vertėmis, centro metodas	21	48,9
Slenkstinės reikšmės viršijimo metodas	21	56,6
Atstumo tarp maksimumų metodas	24	36,1
Mažiausio kvadratų skirtumo metodas	19	139,5
Mažiausio skirtumo metodas	19	16,4
Vienodų plotų metodas	44	3,5
Figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas	23,5	67,5
Suderinto filtro dažnių srityje metodas	19	0,82

Iš anksčiau pateiktos lentelės matome, kad daugiausiai resursų laikine prasme reikalauja masės centro metodas bei mažiausių kvadratų skirtumo metodas. Tai ganėtinai įdomu, mikrovaldiklyje šių metodų įgyvendinimo trukmė ženkliai skiriasi ir vienas lėčiausių metodų mikrovaldiklyje yra kros-koreliacija. Kodėl tiek daug užtrunkama apskaičiuoti rezultatą naudojant mažiausio kvadratų

skirtumo metodą taip pat neaišku, nes metodas turėtų būti greitas. Taip pat matome, kad greičiausias metodas iš visų be jokios konkurencijos yra suderinto filtro dažnių ašyje metodas. Antroje vietoje yra vienodų plotų metodas bei mažiausio skirtumo metodas. Abu šie metodai duoda tokį patį rezultatą, kaip ir kros-koreliacijos metodas. Reikia pabrėžti, kad kros-koreliacijos metodas imamas kaip atraminis, su kuriuo lyginami visi kiti rezultatai.

Literatūroje retai sutinkamas masių centrų metodas vėlavimui tarp signalų nustatyti teoriškai yra labai greitai jo skaičiavimo intensyvumas yra žymiai mažesnis nei kros-koreliacijos metodo, todėl panaudojant jį turėtų būti sutaupoma daug mikrovaldiklio resursų. Kodėl „Matlab“ aplinkoje šio metodo skaičiavimas užtrunka taip ilgai – neaišku. Teoriškai šis metodas turėtų būti žymiai greitesnis už kros-koreliaciją.

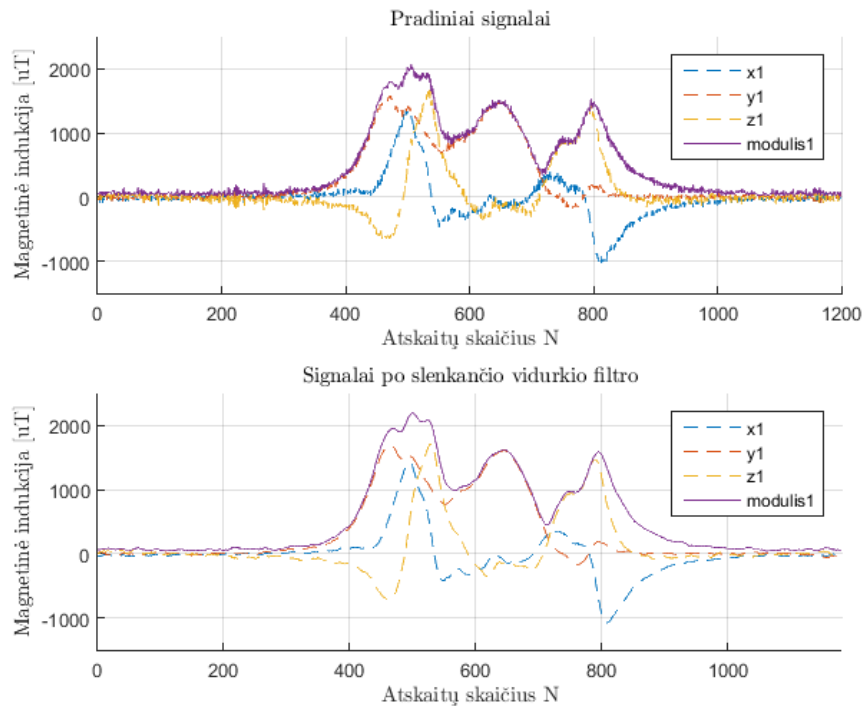
Tokie metodai kaip figūros, apribotos slenkstinėmis vertėmis, slenkstinės reikšmės viršijimo metodas, atstumo tarp maksimumo metodas yra labai paprasti ir greitai apskaičiuojami, jiems nereikia daug mikrovaldiklio skaičiavimo resursų, ko negalima pasakyti apie kros-koreliacijos ir masių centrų metodą ar suderinto filtro dažnių ašyje metodą.

Mažiausio kvadratų skirtumo ir mažiausio skirtumo metodas pasirodė kaip itin gera alternatyva kros-koreliacijai. Teoriškai šis metodas turėtų duoti tikslų skaičiavimo rezultatą netgi esant gan triukšmingiems signalams, o pasirinkus tam tikros eilės slenkantį vidurkį rezultatai turėtų būti dar tikslesni. Plačiau slenkančio vidurkio įtaką metodų rezultatams panagrinėsime sekančiame skyrelyje.

Dalinės kros-koreliacijos metodo skaičiavimų rezultatai plačiau aptarti skyrelyje „Dalinės kros-koreliacijos skaičiavimas“.

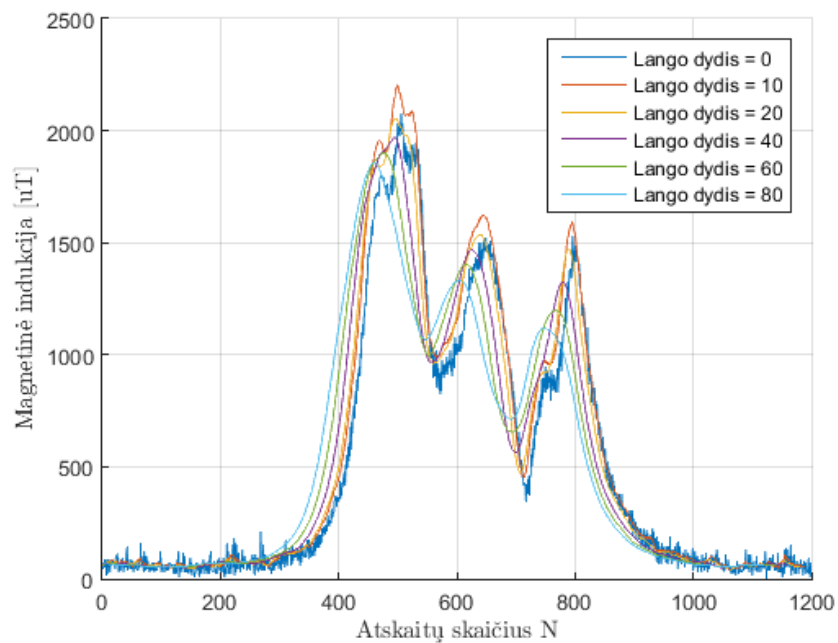
2.14 Vidurkinimo įtaka skaičiavimų rezultatams

Šiame skyrelyje trumpai aptariama slenkančio vidurkio įtaka matavimo rezultatams. Slenkantis vidurkis yra vienas paprasčiausių ir lengviausiai įgyvendinamų būdų signalui filtruoti mikrovaldiklyje. Toks filtravimo metodas pašalina iš pradinių signalų triukšmus, kas leidžia pasiekti tikslesnius rezultatus ieškant vėlinimo tarp dviejų signalų. Žemiau pateikiamas signalų vaizdas prieš ir po slenkančio vidurkio filtro:



35 pav. Viršuje originalūs signalai, apačioje po slenkančio vidurkio, kai vidurkinimo langas yra 10 atskaitų

Iš aukščiau pateikto paveikslo matome, kad slenkantis vidurkis pakeičia signalo formą. Taigi reikia parinkti optimalų slenkančio vidurkio lango dydį, kad pašalintume iš signalų nereikalingus triukšmus. Žemiau pateikiamas vaizdas, kaip pasikeičia vieno iš signalų modulis pritaikius skirtingus slenkančio vidurkio lango dydžius.



36 pav. Slenkančio vidurkio lango dydžio įtaka signalo modulio formai

Iš anksčiau pateikto paveikslo matome, kokią įtaką signalo formai daro slenkančio vidurkio lango didinimas – iš signalo pašalinamos fluktuacijos, o pikinės reikšmės tampa ne tokios aštrios, signalas nesikeičia taip greitai. Žemiau pateikiama lentelė analogiška Lentelei 2, kurioje pateikti visų algoritmų testavimo rezultatai, kai buvo keičiamas slenkančio vidurkio langas.

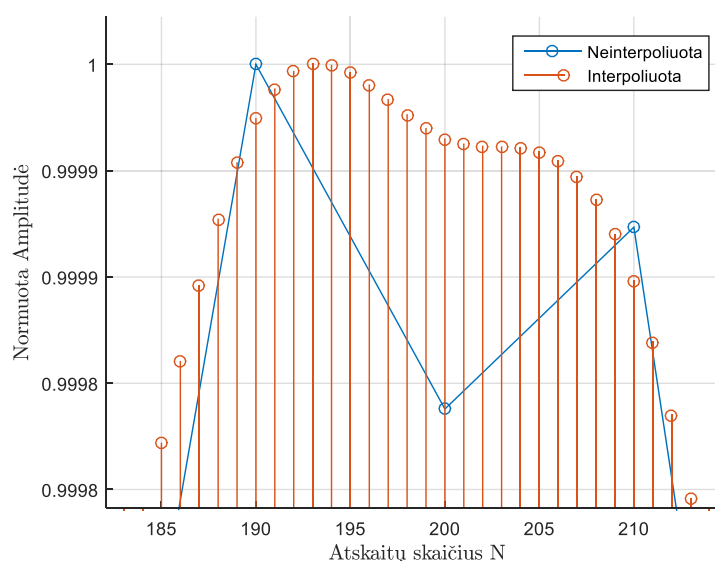
3 lentelė. Slenkančio vidurkio lango įtaką vėlinimo tarp dviejų signalų radimo būdams

Slenkančio vidurkio lango dydis Metodo pavadinimas	Vėlinimas N (atskaitų skaičius)					
	0	10	20	40	60	80
Kros-koreliacija	19	20	20	20	20	20
Masės centro metodas	16,6	16,7	16,8	17,1	17,1	17,1
Figūros, apribotos slenkstinėmis vertėmis, centro metodas	21	21	21,5	22,5	22,5	21,5
Slenkstinės reikšmės viršijimo metodas	21	21	21	21	22	22
Atstumo tarp maksimumų metodas	24	23	22	21	21	21
Mažiausio kvadratų skirtumo metodas	19	20	20	20	20	20
Mažiausio skirtumo metodas	19	20	20	20	20	20
Vienodų plotų metodas	44	40	38	40	39	40
Figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas	23,5	23	21	20,5	20	20
Suderinto filtro dažnių srityje metodas	19	20	20	20	20	20

Iš anksčiau pateiktos lentelės matome, kad nėra prasmės naudoti didesnę lango dydį nei 20 atskaitų, nes daugeliu atveju rezultatai nuo šios reikšmės gaunasi vienodi, nepriklausomai kiek didėja lango dydis. Taigi 20 atskaitų dydis galėtų būti kaip optimalus, siekiant pašalinti iš signalų triukšmus. Matome, kad visais atvejais suderinto filtro dažnių srityje metodas, mažiausio kvadratų skirtumo ir mažiausio skirtumo metodai bei kros-koreliacijos metodai duodavo vienodą rezultatą. Taip pat matome, kad esant dideliame lango dydžiui paprasčiausias figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas taip pat duodavo vienodą rezultatą, kaip ir aukščiau paminėti skaičiavimo būdai. Šie metodai vėliau bus įgyvendinti mikrovaldiklyje ir palyginsime, kaip skaičiavimo laikas skiriasi realioje aplinkoje nuo skaičiavimo laiko „Matlab“ programiniu paketu.

2.15 Rezultatų patikslinimo būdai

Aukščiau aptartus metodus galima naudoti kai reikia surasti vėlinimą tarp dviejų magnetinių signalūrų, kurias sukuria virš magnetinio lauko jutiklio pravažiuojanti transporto priemonė. Problema yra ta, kad atstumas tarp magnetinio lauko jutiklių yra tik 30 centimetrų, taigi jeigu transporto priemonė pravažiuoja dideliu greičiu, mes negalime jo tiksliai išmatuoti dėl per mažo diskretizavimo dažnio. Vienas iš būdų kaip galima padidinti matavimo tikslumą yra padidinti signalo diskretizavimo dažnį, kad turėtume daugiau atskaitų, tačiau tai neįmanoma, dauguma šiuolaikinių skaitmeninių magnetinio lauko jutiklių negali pasiekti didesnio diskretizavimo dažnio nei 1 kHz. Tuomet vienas iš būdų kaip galima gauti didesnę rezoliuciją yra dirbtiniu būdu pridėti daugiau atskaitų į signalą (interpoliuoti). Tuomet signalai turės daugiau atskaitų ir bus galima gauti didesnę rezoliuciją. Žemiau pateikiamas pavyzdys, kaip pasikeičia kros-koreliacijos skaičiavimo rezultatas kai rezultatui apskaičiuoti signalai yra interpoliuojami.



37 pav. Normuotos kros-koreliacinės funkcijos, viena gauta su interpoliuotais signalais

Aukščiau pateiktame paveiksle mėlynas signalas yra kros-koreliacinės funkcijos viršūnė, gauta naudojant dvi magnetines signalūras. Raudonas signalas yra taip pat kros-koreliacinės funkcijos viršūnė, tik šį kartą jai skaičiuoti buvo panaudoti 10 kartų interpoliuoti signalai. Tam, kad vizualiai galėtume pamatyti, kaip pasikeičia rezultatas, sunormuojame abi kros-koreliacines funkcijas pagal amplitudę. Iš gauto pavyzdžio matome, kad jeigu signalų neinterpoliuojame, vėlinimas tarp signalų šiuo atveju yra lygus 19. Tuo tarpu panaudojus 10 kartų interpoliuotus signalus gauname, kad vėlinimas tarp signalų yra 19.3 (reikia surasti kros-koreliacinės funkcijos indeksą, kuriame funkcijos reikšmė lygi 1). Aukščiau pateiktas paveikslas yra grynai tik iliustracinio pobūdžio, realiame

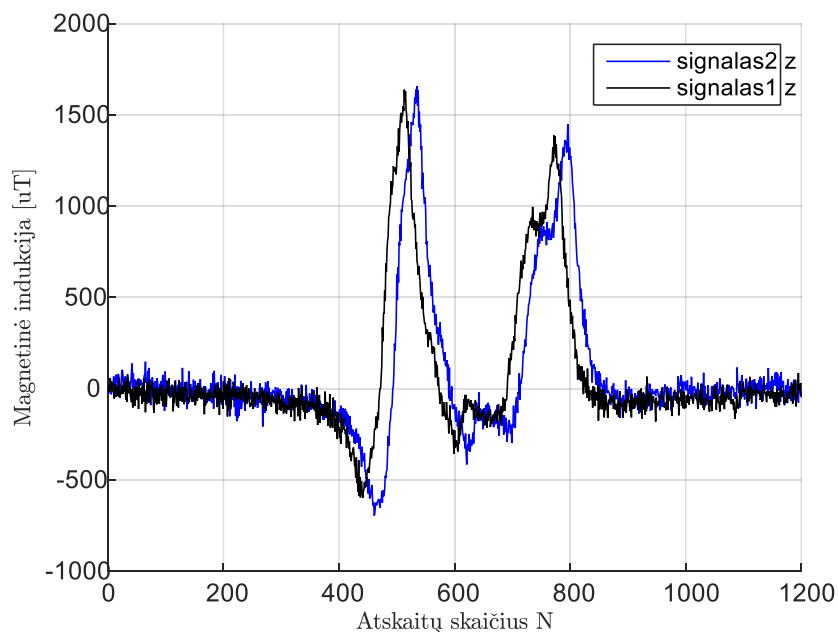
mikrovaldiklyje signalai nebūtų normuojami, nes mikrovaldiklyje laikine prasme tikslingiau atlikti skaičiavimus su sveikais skaičiais, o ne su slankaus kablelio tipo skaičiais.

Pagrindinis aukščiau aprašyto metodo trūkumas yra tas, kad interpoliuotus signalus reikia saugoti mikrovaldiklio atmintyje. Šiuo atveju, kadangi kros-koreliacijai apskaičiuoti reikalingas visas signalas, galime paskaičiuoti, kiek kartų padidėtų reikalingos atminties kiekis mikrovaldiklyje. Pavyzdžiui, turime atmintyje saugoti dvi signatūras, kurios turi po tūkstantį atskaitų. Vienai atskaitai saugoti reikalingi du baitai. Vadinasi saugoti dvi signatūras po tūkstantį atskaitų reikės keturių tūkstančių baitų arba 4 kB. Jeigu interpoliuojame signalus dešimt kartų, vadinasi tuos pačius signalus saugoti mikrovaldiklio atmintyje jau reikės 40 kB. Tai dideli skaičiai, tokį atminties kiekį turi tik galingesni ir brangesni mikrovaldikliai, o jeigu reikėtų signalus saugoti išorinėje atmintyje tai sugaištume daug laiko. Vadinasi šis metodas padidina rezultato tikslumą, tačiau tam reikėtų labai daug papildomų mikrovaldiklio resursų. Galbūt galima signalą interpoliuoti du ar keturis kartus, tuomet vis tiek turėtume tikslesnį rezultatą nei originaliu atveju, ir nenaudotume tiek daug mikrovaldiklio atminties, tačiau šio skyrelio tikslas parodyti, kad matavimo tikslumą galima padidinti.

Šį metodą galima pritaikyti bet kokiam vėlinimo tarp dviejų signalų radimo būdai. Vienas iš būdų kaip ateityje bus bandoma padidinti matavimo tikslumą tai pritaikyti interpoliaciją slenkstinės vertės viršijimo metodams. Kai naudojami šie skaičiavimo būdai, nereikia saugoti visų signalų mikrovaldiklio atmintyje. Pavyzdžiui, galima surasti taškus, kuriuose signalai kerta slenkstinę reikšmę. Tuomet užsiduoti tam tikrą intervalą ir tik tame siaurame intervale interpoliuoti signalus ir taip apskaičiuoti naują vėlinimo reikšmę. Toks metodas reikalautų atmintyje laikyti vos keliasdešimt atskaitų ir būtų žymiai greitesnis lyginant su skaičiavimo būdais, kuriems reikalinga saugoti visą signalą (kros-koreliacija, mažiausių kvadratų skaičiavimas, vienodų plotų metodas ir t.t.).

2.16 Vėlinimo tarp dviejų signalų skaičiavimo būdų pritaikymas Z ašiai

Kadangi galutinis šio darbo tikslas yra kaip įmanoma greičiau ir tiksliau surasti vėlinimą tarp dviejų signalų, verta pabandyti aukščiau aprašytus metodus pritaikyti tik magnetinio lauko jutiklių Z ašiai, mat šį komponentė yra statmena transporto priemonės judėjimo krypties. Žemiau pateikiamas paveikslas, kaip atrodo aukščiau pavyzdžiuose naudotų dviejų signalų modulio Z komponentės.



38 pav. Dviejų signalų modulių Z komponentės

Žemiau esančioje lentelėje pateikiamas visų vėlinimo tarp dviejų signalų radimo būdų skaičiavimo rezultatas, kai skaičiavimai atliekami su signalų moduliais, ir kai su modulio Z komponente.

4 lentelė. Metodų taikymas signalų moduliams ir Z ašims.

Metodo pavadinimas	Vėlinimas N (atskaitų skaičius)		
	Naudojami moduliai	Naudojamos Z komponentės	Naudojamos Z komponentės ir slenkantis vidurkis(Lango dydis = 20)
Kros-koreliacija	19	21	21
Masės centro metodas	16,61	49,84	50,55
Figūros, apribotos slenkstinėmis vertėmis, centro metodas	21	21	22
Slenkstinės reikšmės viršijimo metodas	21	23	22
Atstumo tarp maksimumų metodas	24	22	22
Mažiausio kvadratų skirtumo metodas	19	21	21
Mažiausio skirtumo metodas	19	21	21
Vienodų plotų metodas	44	39	44

4 lentelės tęsinys

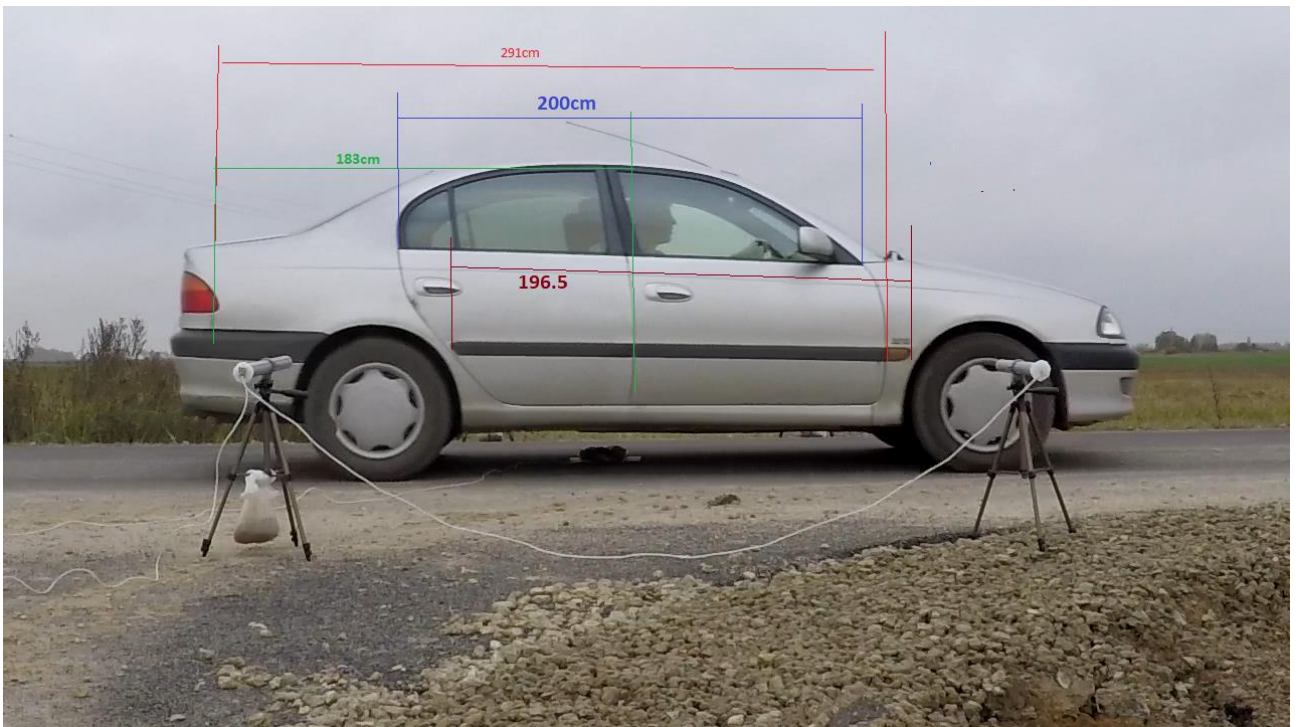
Figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas	23.5	24	22
Suderinto filtro dažnių srityje metodas	19	21	21

Iš pateiktos lentelės matome, kad masės centro skaičiavimas duoda visiškai kitokį rezultatą skaičiuojant su Z komponente. To priežastis, kad Z komponentė turi ir neigiamas reikšmes, dėl jų iškraipomas masės centras (skaičiavimai turėtų būti atliekami tik su signalais, kurie neturi neigiamų atskaitų). Nors kiti skaičiavimo būdai, tokie kaip Kros-koreliacija, mažiausių kvadratų skirtumo ar suderinto filtro dažnių srityje duoda kitokį rezultatą nei naudojant modulius, tačiau Z ašyje visi šie skaičiavimo būdai duoda vienodą rezultatą. Iš to galime padaryti išvadą, kad visi šie metodai (nepriklausomai nuo to kokie bus signalai) duoda vienodą skaičiavimo rezultatą, vadinasi betkurio iš šių skaičiavimo būdų rezultatas gali būti laikomas kaip vėlinimas atskaitomis tarp signalų.

Pritaikius slenkantį vidurkį, kurio lango dydis yra lygus 20, matome, kad kai kurių skaičiavimo būdų rezultatas pasikeičia, tačiau pagrindiniais laikomi skaičiavimo būdai (kros-koreliacija, mažiausių skirtumų, suderinto filtro metodas) duoda tokius pačius rezultatus, tačiau jie skiriasi nuo rezultatų, kurie gauti naudojant tik signalų modulius. Todėl šiuo atveju nėra aišku, skaičiavimai gaunami tikslesni ar su moduliais, ar su z komponentėmis. Teoriškai, rezultatai turėtų būti tikslesni naudojant signalų modulius, nes jų formai įtaką turi ir x , ir y komponentės. Reikėtų išbandyti visus skaičiavimo būdus abejais atvejais su daug transporto priemonių, kad galėtume aiškiai teigti, kuriuo atveju gauti rezultatai yra tikslesni.

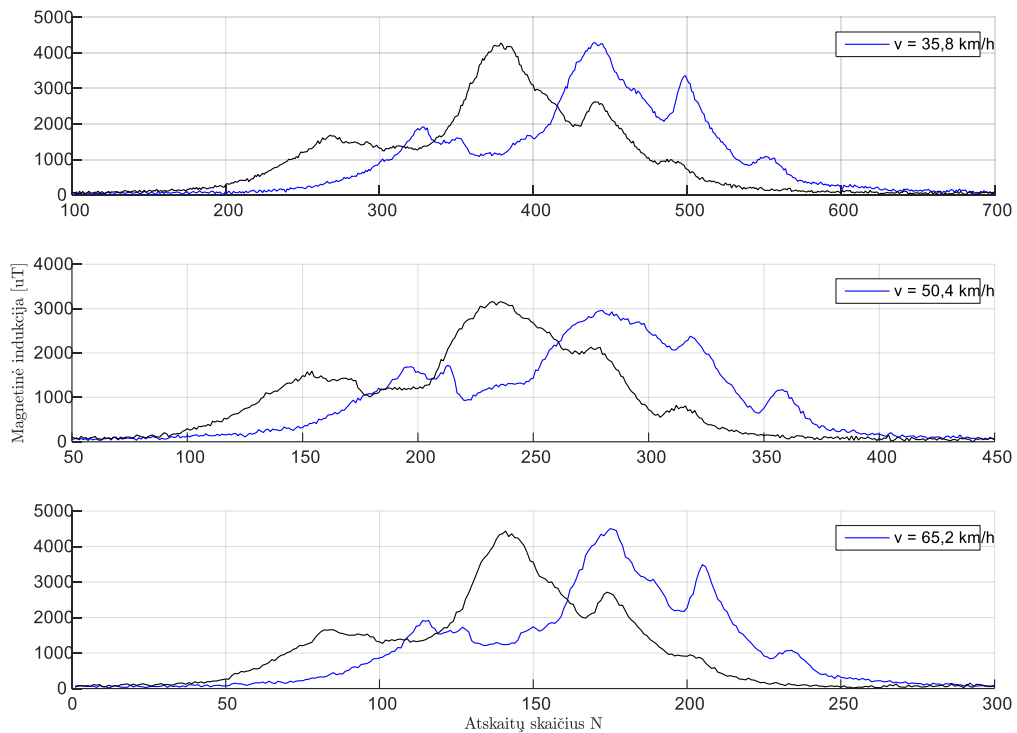
3. EKSPERIMENTINIAI REZULTATAI

Anksčiau aptarti vėlinimo tarp dviejų signalų radimo būdai išbandyti su signatūromis, kurios sukūrė žinomu greičiu virš magnetinio lauko jutiklių pravažiavusi transporto priemonė. Šio eksperimento metu naudota transporto priemonė „Toyota Avensis“. Pravažiuojanti virš magnetinio lauko jutiklių transporto priemonė buvo filmuojama „GoPro“ kamera, kurios kadro skaičius per sekundę lygus 240. Vėliau iš filmuotos medžiagos vaizdų apdorojimo pagalba buvo nustatytas transporto priemonės greitis, kuris naudojamas kaip tikrasis transporto priemonės greitis. Transporto priemonė virš jutiklių pravažiavo trimis skirtingais greičiais. Taip pat transporto greitis buvo fiksuojamas ir optiniu būdu – tam tikru atstumu vienas nuo kito stovėjo du foto diodai, į kuriuos buvo nutaikyti du atskiri šviesos šaltiniai. Kai pro juos pravažiuoja transporto priemonė, šviesos kelias uždengiamas ir foto diodo išėjime pasikeičia lygis. Tokiu būdu išmatuojamas laiko tarpas, kol lygis foto diodų išėjime pasidarė vienodas, pasikeitus pirmajam. Žemiau pateikiama nuotrauka eksperimento padaryta eksperimento metu.



39 pav. Eksperimentas realiomis sąlygomis su „Toyota Avensis“ automobiliu

Žemiau pateikiamas paveikslas su trimis skirtingomis magnetinėmis signatūromis, kurios gautos esant skirtingiems greičiams (prie signatūros prirašytas greitis yra gautas iš GoPro vaizdo kameros).



40 pav. Magnetinės signalų esant trims skirtingiems greičiams

Žemiau esančioje lentelėje pateikiamas išmatuotas transporto priemonės greitis dviem metodais (vaizdo kamera ir optiniu būdu).

5 lentelė. Transporto priemonės greitis užfiksuotas dviem skirtingais būdais

Bandymo numeris	Greitis $\frac{km}{h}$	
	„GoPro“ vaizdo kamera	Optinis būdas
1	35,8	36,5
2	50,4	49,9
3	65,2	65,5

Iš aukščiau pateiktos lentelės matome, kad tiek vaizdo kamera, tiek optiniu būdu gauti rezultatai yra labai panašūs. Šiuo atveju kaip tikrąjį transporto priemonės greitį priimame greitį, kuris gautas su „GoPro“ vaizdo kamera. Žemiau esančiose lentelėse pateikiamas apskaičiuotas transporto priemonės greitis naudojant visus aukščiau aptartus vėlinimo tarp dviejų signalų radimo metodus. Taip pat pateikiamas transporto priemonės greitis pritaikius skirtingus slenkančio vidurkio lango dydžius.

6 lentelė. Transporto priemonės greitis nustatytas iš magnetinės signalūros pirmuoju bandymu

Bandymas Nr 1. $v = 35,8 \frac{km}{h}$	Greitis km/h			
	Naudojami Metodai	Naudojami moduliai	Naudojamos Z komponentės	Naudojami Moduliai ir slenkantis vidurkis(Lango dydis = 20)
Kros-koreliacija	36	37,2	35,4	37,2
Masės centro metodas	35,2	52,6	35,3	49,9
Figūros, apribotos slenkstinėmis vertėmis, centro metodas	34,3	32,7	34,8	42,3
Slenkstinės reikšmės viršijimo metodas	35,4	36	35,4	42,3
Atstumo tarp maksimumų metodas	35,4	36,6	35,4	38,6
Mažiausio kvadratų skirtumo metodas	36	37,2	35,4	37,2
Mažiausio skirtumo metodas	35,4	37,2	34,8	37,9
Vienodų plotų metodas	17,4	-	17,1	-
Figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas	35,1	130,9	33,8	42,3
Suderinto filtro dažnių srityje metodas	36	37,2	35,4	37,2

7 lentelė. Transporto priemonės greitis nustatytas iš magnetinės signalūros antruoju bandymu

Bandymas Nr 2. $v = 50,4 \frac{km}{h}$	Greitis km/h			
	Naudojami Metodai	Naudojami moduliai	Naudojamos Z komponentės	Naudojami Moduliai ir slenkantis vidurkis(Lango dydis = 20)
Kros-koreliacija	49	51,4	49	51,4
Masės centro metodas	53,2	24,4	53	25,6
Figūros, apribotos slenkstinėmis vertėmis, centro metodas	9,6	166,2	130,9	108
Slenkstinės reikšmės viršijimo metodas	67,5	55,4	60	60
Atstumo tarp maksimumų metodas	45	61,7	47	52,7

7 lentelės tęsinys

Mažiausio kvadratų skirtumo metodas	49,1	51,4	48	51,4
Mažiausio skirtumo metodas	49,1	52,7	48	54
Vienodų plotų metodas	34,3	-	32,7	-
Figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas	50,8	83,1	50,2	70,8
Suderinto filtro dažnių srityje metodas	49,1	51,4	48	51,4

8 lentelė. Transporto priemonės greitis nustatytas iš magnetinės signatūros trečiuoju bandymu

Bandymas Nr 3. $v = 65,2 \text{ km/h}$	Greitis $\frac{km}{h}$			
	Naudojami Moduliai	Naudojamos Z komponentės	Naudojami Moduliai ir slenkantis vidurkis(Lango dydis = 20)	Naudojamos Z komponentės ir slenkantis vidurkis(Lango dydis = 20)
Kros-koreliacija	67,5	69,7	65,5	69,7
Masės centro metodas	69,1	74,7	67,7	72,4
Figūros, apribotos slenkstinėmis vertėmis, centro metodas	240	154,3	166,1	160
Slenkstinės reikšmės viršijimo metodas	80	74,5	77,1	83,1
Atstumo tarp maksimumų metodas	63,5	74,5	65,5	69,7
Mažiausio kvadratų skirtumo metodas	67,5	69,7	65,5	69,7
Mažiausio skirtumo metodas	67,5	69,7	65,5	69,7
Vienodų plotų metodas	50,2	-	42,4	-
Figūros, apribotos slenkstinėmis reikšmėmis vidurkio metodas	73,2	60	67,5	80
Suderinto filtro dažnių srityje metodas	67,5	69,7	65,5	69,7

Iš aukščiau pateiktų lentelių galima padaryti keletą išvadų. Pirmoji tai ta, kad jeigu skaičiavimas naudojamos Z signalo komponentės, tai rezultatai yra gerokai blogesni, nei, kad naudojant signalų modulius. Nesvarbu ar naudojami nefiltruoti duomenys, ar perleisti per slenkančio vidurkio filtrą, rezultatai vis tiek gaunami su žymiai didesne paklaida, negu naudojant signalų modulius. Taigi galima

konstatuoti, kad nepavyks gauti norimo skaičiavimo tikslumo naudojant tik magnetinio lauko sensoriaus Z komponentes.

Taip pat matome, kad kai skaičiavimas naudojami signalų moduliai, slenkančio vidurkio naudojimas ne visais atvejais pagerina situaciją, tačiau jį naudoti rekomenduoja visi analizuoti literatūros šaltiniai. Tai yra būtina norint pašalinti visus atsitiktinius triukšmus, tačiau atkreiptinas dėmesys, jog kartais dėl to rezultatas pasikeičia nebūtinai į gerąją pusę.

Dar vienas dalykas į kurį reikėtų atkreipti dėmesį yra tas, kad kai kurie metodai yra visiškai netikslūs, pavyzdžiui vienodų plotų skaičiavimas, figūros, apribotos slenkstinėmis vertėmis centro skaičiavimas, masės centro skaičiavimas taip pat duoda netikslų rezultatą, todėl eksperimentinėje dalyje šie metodai net netestuojami kaip visiškai netikslūs ir neturintys prasmės. Pavyzdžiui, kai tikrasis transporto priemonės greitis yra $65,2 \text{ km/h}$, tai skaičiuojant vienodus plotus ir naudojant nefiltruotus signalus gaunamas $50,2 \text{ km/h}$ greitis. Ieškant figūros, apribotos slenkstinėmis vertėmis centro, gaunamas transporto priemonės 240 km/h . Ieškant figūros, apribotos slenkstinėmis vertėmis vidurio, gaunamas transporto priemonės greitis $73,2 \text{ km/h}$. Be to taikant šiuos metodus jų rezultatams didelę įtaką turi slenkančio vidurkio naudojimas, skaičiavimas naudojant Z komponentes o ne modulius, dėl to nėra prasmės šių metodų skaičiavimą išbandyti realioje sistemoje. Taip pat matome, kad mažiausio kvadratų skirtumo skaičiavimas, mažiausio skirtumo skaičiavimas, kros-koreliacijos ir suderinto filtro metodas beveik visada duoda vienodus rezultatus. Iš to taip pat galime daryti prielaidą, kad šie metodai tiksliai apskaičiuoja vėlinimą tarp signalų jeigu naudojami signalų moduliai, nes visais atvejais gaunamas vienodas rezultatas (jis nebūtinai yra teisingas, tačiau gautas rezultatas byloja, kad vėlinimas tarp signalų yra būtent toks).

3.1 Vėlinimo tarp signalų skaičiavimas mikrovaldiklyje

Aukščiau aprašyto bandymo metu buvo gautos trys skirtingos signalų poros esant skirtingiems automobilio greičiams. Įgyvendinti mikrovaldiklyje aukščiau aprašytus metodus buvo pasirinkta signalų pora, kuri gauta esant didžiausiam greičiui. Taip pasirinkta dėl to, kad kuo didesnis transporto priemonės greitis, tuo didesnės paklaidos jeigu netaikomi jokie greičio patikslinimo metodai. Skaičiavimams panaudota „STM32F767 Nucleo“ maketinė plokštė, kurioje yra STM32F767ZI mikrovaldiklis. Šio mikrovaldiklio taktinis dažnis 216 MHz.

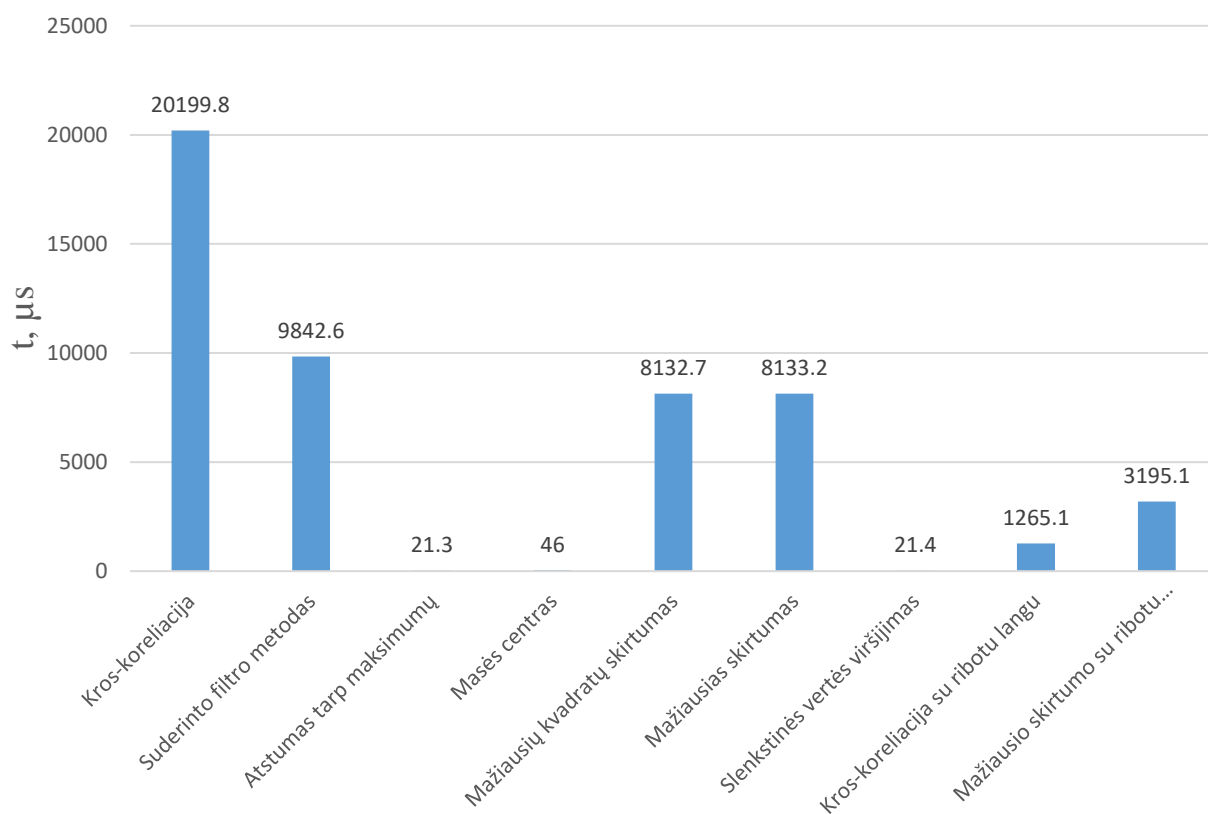
Žemiau esančioje lentelėje pateikiamas testuotų metodų sąrašas mikrovaldiklyje ir gautas transporto priemonės greitis, bei kiek laiko užtruko mikrovaldiklis kol suskaičiavo galutinį rezultatą (laikoma, kad signalūros jau yra pilnai paruoštos apdorojimui. Realioje sistemoje šis laikas būtų didesnis dėl to, kad reiktų realiu laiku aptikti signalūros pradžią, pabaigą, galbūt nufiltruoti ir

patalpinti į mikrovaldiklio atmintį prieš pradėdant tolimesnį apdorojimą). Trukmė, kiek laiko užtruko apskaičiuoti rezultatą buvo matuojama dviem būdais: mikrovaldiklio vidiniu laikmačiu, kurio taktinis dažnis 108 MHz ir Osciloskopu „Yokogawa DLM2054 (2.5 Gs/s)“. Matuojant osciloskopu, kaip laiko atskaitos buvo imamos 90% kylančio ir krintančio signalo fronto reikšmės. Signalo ilgis šiuo atveju yra lygus 300 atskaitų, mašinos tikrasis greitis $v = 65,2 \frac{km}{h}$. Skaičiavimams naudoti signalai nebuvo praleisti pro slenkančio vidurkio filtrą.

9 lentelė. STM32F767ZI mikrovaldiklio skaičiavimų rezultatai.

Skaičiavimo būdas	$T(Osciloskopu), \mu s$	$T(MCU\ Taimeriu), \mu s$	Greitis, km/h
Kros-koreliacija	20199,8	20182,6	67,5
Suderinto filtro metodas	9842,6	9842,6	67,5
Atstumas tarp maksimumų	21,3	21,0	63,5
Masės centras	46,0	45,7	69,7
Mažiausio kvadratų skirtumo skaičiavimas	8132,7	8132,4	67,5
Mažiausio skirtumo skaičiavimas	8133,2	8133,7	67,5
Slenkstinės vertės viršijimas	21,4	21,1	67,5
Kros-koreliacija su ribotu langu	1265,1	1264,8	67,5
Mažiausio skirtumo su ribotu langu	3195,1	3194,6	67,5

Toliau esančiame paveiksle pateikiama kiekvieno metodo skaičiavimo trukmė mikrovaldiklyje.



41 pav. Visų testuotų metodų skaičiavimo trukmė mikrovaldiklyje

Visi aukščiau esančioje lentelėje pateikti rezultatai buvo gauti išjungus visas mikrovaldiklio kodo optimizacijas. Matome, kad daugiausia laiko gauti rezultatui reikia skaičiuojant kros-koreliaciją, tačiau kaip jau minėjome ankstesniuose skyriuose, skaičiuoti kros-koreliacija per visą signalo ilgį yra visiškai netikslinga. Štai pavyzdžiui kros-koreliacijos metodas su ribotu langu yra įgyvendinamas žymiai greičiau ($20199,8 \mu s$ ir $1265,1 \mu s$). Šio skaičiavimo būdo esmė, kad kros-koreliacija skaičiuojama tik siauroje signalo dalyje, kurioje manoma jog yra rezultatas, o apytikslį rezultatą galime gauti iš bet kurio paprasto skaičiavimo būdo (šiuo atveju buvo panaudotas atstumas tarp maksimumų). Toks rezultato gavimo būdas yra daug žadantis, nes rezultatas apskaičiuojamas ganėtinai greitai, o rezultato tikslumas nuo to nei kiek nepasikeičia. Tokiu pačiu principu galima naudoti mažiausio skirtumo su ribotu langu skaičiavimo būdą, tačiau rezultatai skaičiavimo prasme yra šiek prastesni nei kros-koreliacijos skaičiavimo atveju. Taip pat šiuo atveju galime palyginti greičio paklaidas.

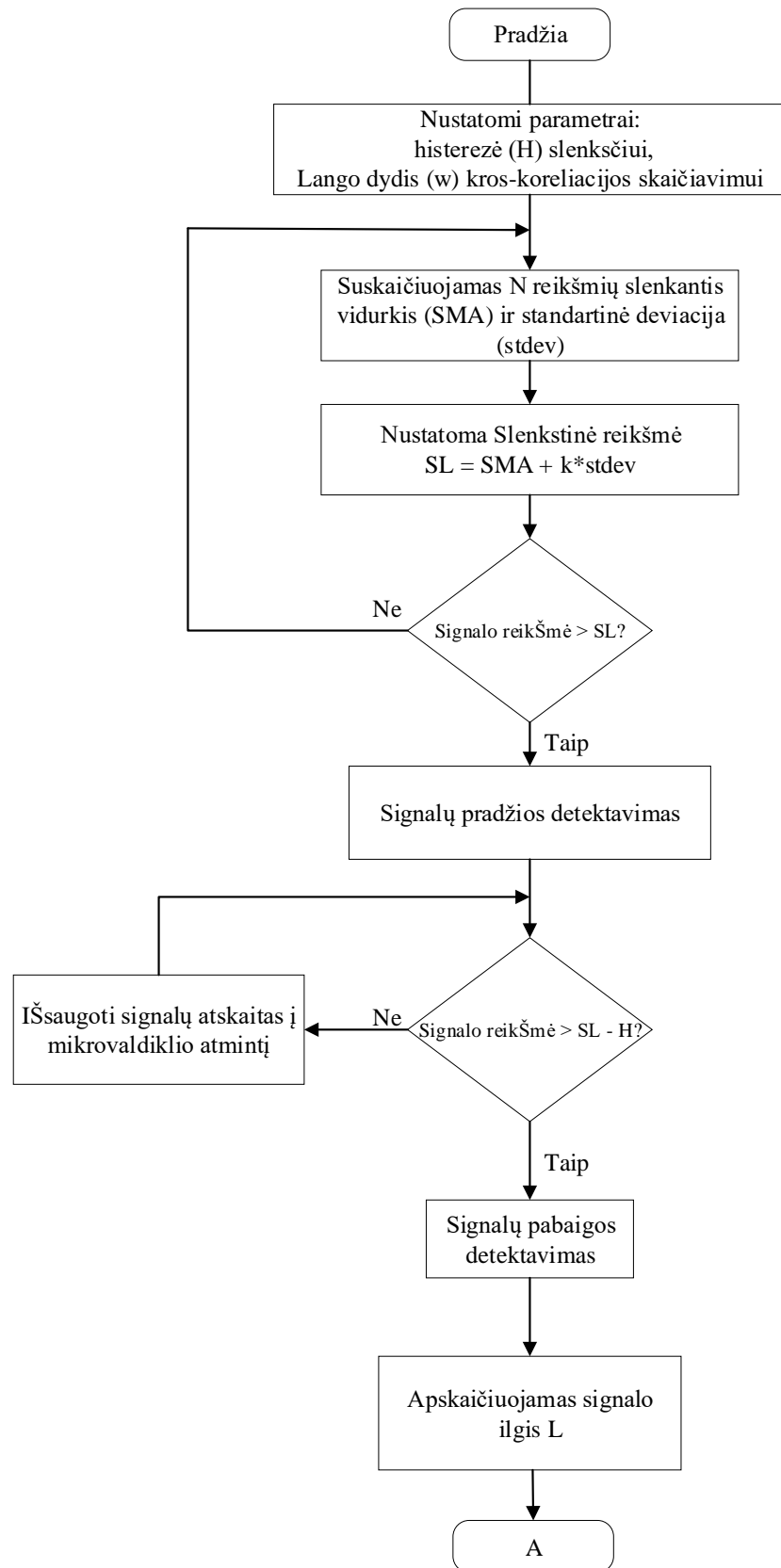
10 lentelė. Eksperimentiniu būdu gautos absoliutinės ir santykinės greičio paklaidos

Skaičiavimo būdas	Absoliutinė paklaida Δv , <i>km/h</i>	Santykinė paklaida %
Kros-koreliacija	2,3	3,4
Suderinto filtro metodas	2,3	3,4
Atstumas tarp maksimumų	1,7	2,6
Masės centras	4,5	6,5
Mažiausio kvadratų skirtumo skaičiavimas	2,3	3,4
Mažiausio skirtumo skaičiavimas	2,3	3,4
Slenkstinės vertės viršijimas	2,3	3,4
Kros-koreliacija su ribotu langu	2,3	3,4
Mažiausio skirtumo su ribotu langu	2,3	3,4

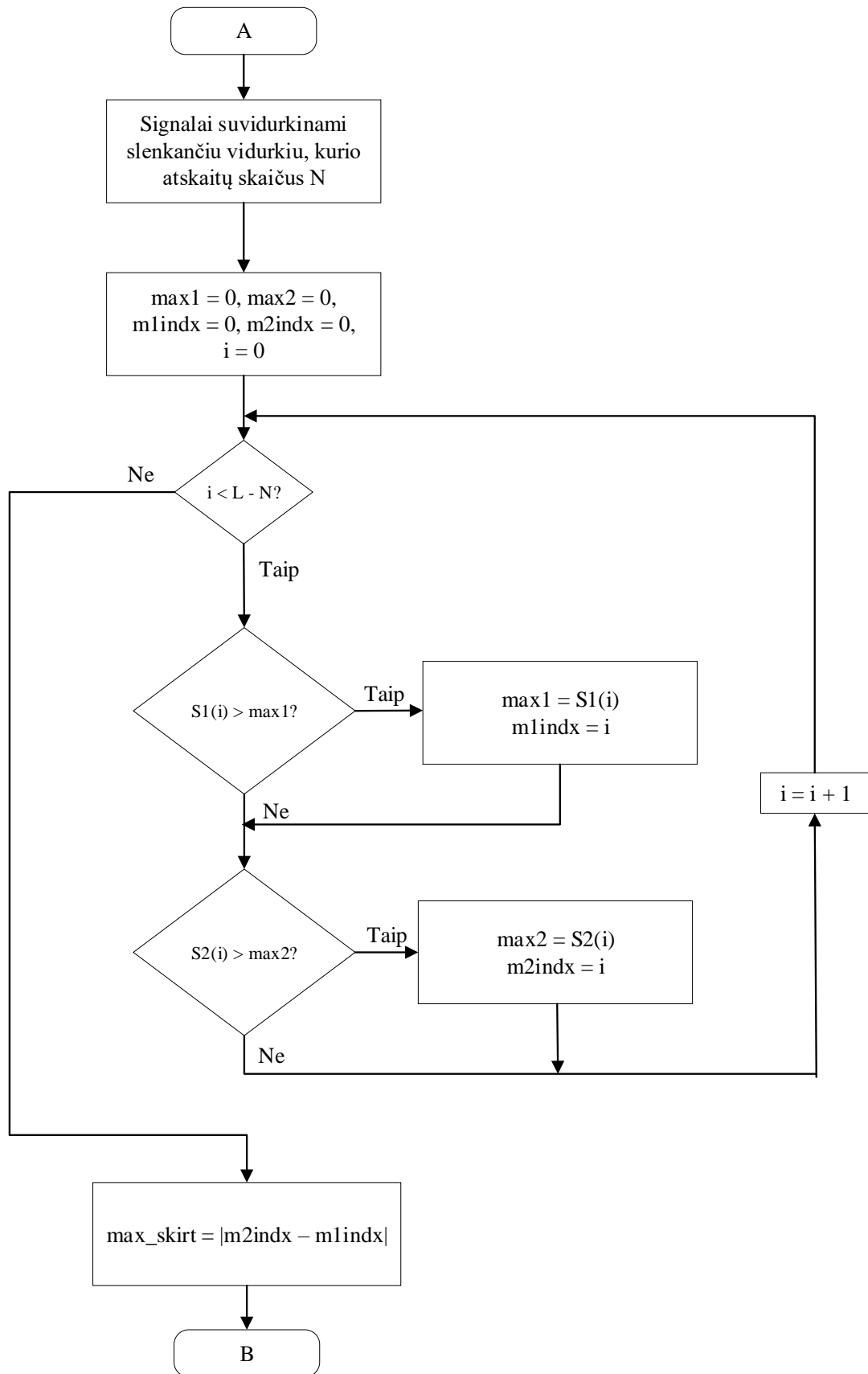
Iš anksčiau pateiktos lentelės matome, kad greičio paklaidos yra gan didelės, kai transporto priemonė važiuoja $65,2 \text{ km/h}$ greičiu. Daugelis šiuolaikinių greičio matavimo sistemų ar radarų greitį matuoja gerokai tiksliau. Be to, tokioje sistemoje matavimo tikslumas mažėja didėjant transporto priemonės greičiui, taigi kuo didesnis greitis, tuo didesnė matavimo paklaida bus gaunama. Tačiau visumoje reikia paminėti, kad matavimo paklaidos net ir esant dideliems greičiams turėtų tilpti į 10%. Aišku tai didelė paklaida, tačiau tuomet galima taikyti greičio patikslinimo metodus. Šio darbo esmė parodyti, kad iš magnetinių signalų apskaičiuoti transporto priemonės greitį yra įmanoma.

Po šių eksperimentų galime padaryti rezultatų apibendrinimą ir pateikti rekomenduojamą naudoti sprendimą. Sistema bet kokiame atveju turėtų naudoti slenkančio vidurkio filtrą, tam, kad pašalinti iš signalų triukšmus ir atsitiktinius artefaktus. Norint pasiekti greičiausią skaičiavimo laiką, reikia taikyti dalinės kros-koreliacijos skaičiavimo būdą. Kaip pagalbinį metodą, kuriuo apytiksliai nustatytume vėlinimą tarp signalų galima naudoti bet kurį greitai apskaičiuojamą metodą: atstumą tarp maksimumų, slenkstinės vertės viršijimą, figūros, apribotos slenkstinėmis reikšmėmis iš abiejų pusių vidurkio skaičiavimą ir t.t. Kros-koreliaciją duoda tikslų matavimo rezultatą ir šį skaičiavimo būdą galima itin pagreitinti naudojant aukščiau aprašytą taktiką.

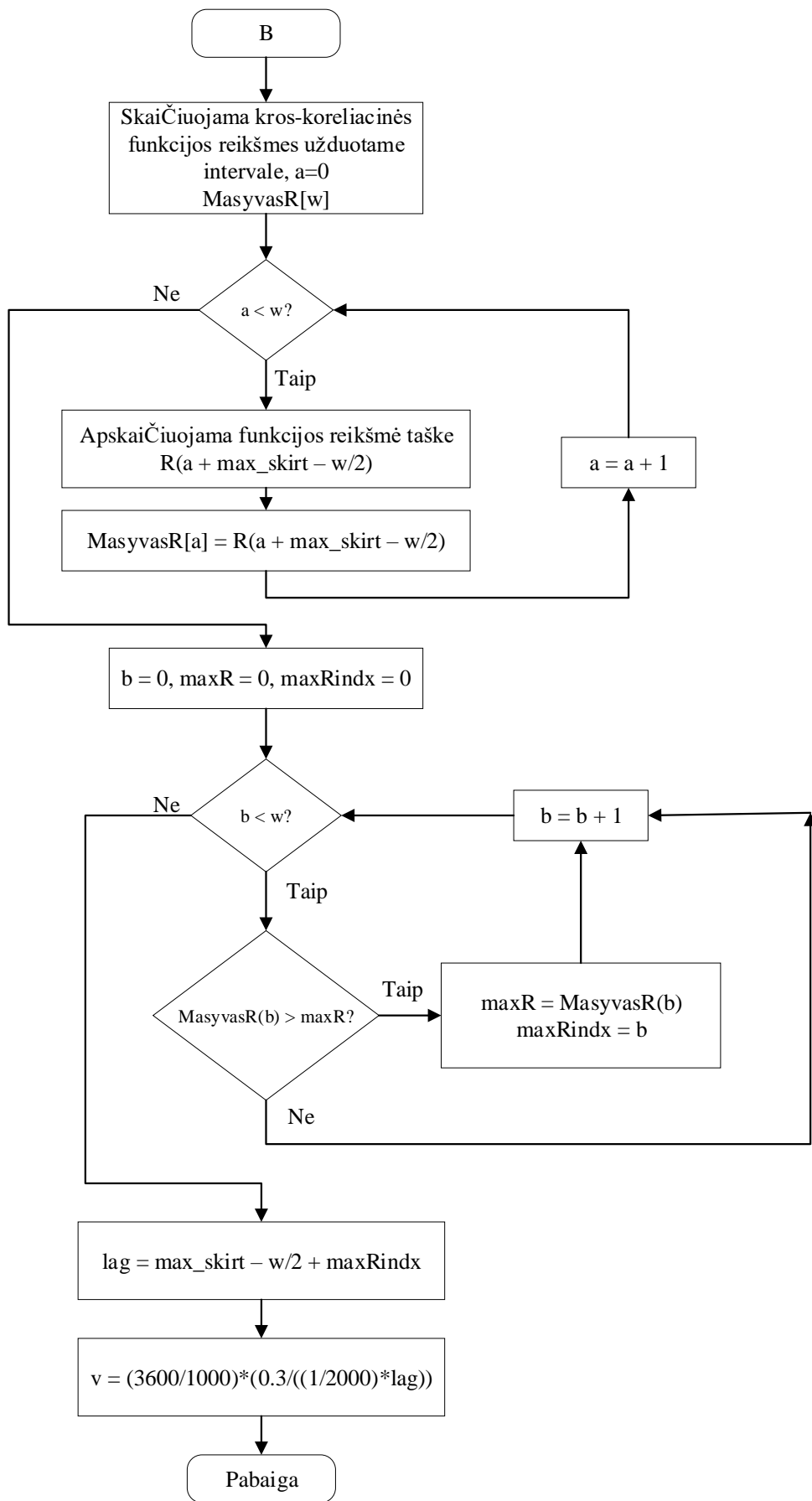
Toliau pateikiamas galutinio, rekomenduojamo naudoti, transporto priemonės greičio nustatymo algoritmas.



a) Pirmoji algoritmo dalis



b) Antroji algoritmo dalis



42 pav. Sukurtas algoritmas transporto priemonės greičiui apskaičiuoti

IŠVADOS IR PASIŪLYMAI

1. Darbe išanalizuoti ir praktiškai išbandyti įvairūs vėlinimo tarp dviejų signalų radimo algoritmai, kai duomenys gaunami iš AMR tipo magnetinio lauko jutiklių. Įvertintas šių skaičiavimo būdų tikslumas ir skaičiavimo trukmė. Tai aktualu siekiant perkelti realų sprendimą į tikrą sistemą, kurioje visus procesus valdys mikrovaldiklis.
2. Daugumos magnetinio lauko jutiklių diskretizavimo dažnis (iki 1 kHz) yra per mažas, kad būtų galima pakankamai tiksliai nustatyti dideliu greičiu važiuojančios transporto priemonės greitį. Tokiu atveju reikia naudoti greičio patikslinimo būdus, vienas iš galimų variantų šiame darbe – signalų interpoliacija. Realaus eksperimento metu signalų interpoliavimas sumažino santykinę paklaidą 5,1 karto (nuo 0,46 % iki 0,09 %) kai transporto priemonės greitis buvo 65,2 km/h ir 1,48 karto (nuo 1,13 % iki 0,76 %), kai transporto priemonės greitis buvo 35,8 km/h.
3. Atliktas eksperimentas realiomis sąlygomis ir įvertinta skaičiavimo trukmė bei transporto priemonės greičio nustatymo tikslumas, naudojant vienuolika vėlinimo tarp dviejų signalų nustatymo būdų. Kros-koreliacija, suderinto filtro metodas, MKS ir MS metodų atvejais santykinė paklaida greičių diapazone nuo 35,8 km/h iki 65,2 km/h gaunama nuo 1,2 % iki 3,4 %. Kitais skaičiavimo būdais gaunamos paklaidos yra didesnės.
4. Darbe įvertinta vidurkinimo įtaka skaičiavimo rezultatams. Naudojant slenkantį vidurkį, kurio lango dydis yra 20 atskaitų ir apskaičiuojant TP greitį diapazone nuo 35,8 km/h iki 65,2 km/h santykinė paklaida gaunama nuo 0,45 % iki 2,80 %.
5. Norint gauti kuo tikslesnius rezultatus, skaičiuojant pravažiuojančios transporto priemonės greitį, tikslinga naudoti magnetinio lauko jutiklių komponentių modulį. Naudojant tik Z komponentę gaunamas mažesnis tikslumas negu naudojant signalų modulius. Kros-koreliacijos skaičiavimo atveju naudojant Z komponentes santykinė matavimo paklaida gaunama 1,91 karto didesnė nei naudojant signalų modulius (6,5 % ir 3,4 %), kai transporto priemonės greitis 65,2 km/h.
6. Rekomenduojama realiomis sąlygomis greičio nustatymui naudoti dalinės kros-koreliacijos skaičiavimo būdą. Kaip pagalbinį metodą rekomenduojama naudoti atstumą tarp maksimumų arba slenkstinės vertės viršijimą. Šiais būdais mikrovaldiklyje skaičiavimai atliekami 950 kartų greičiau nei kros-koreliacija, 460 kartų greičiau nei suderinto filtro metodas ir 380 kartų greičiau nei mažiausių skirtumo skaičiavimas.
7. Naudojant pasiūlytą algoritmą skaičiavimai mikrovaldiklyje atliekami 7-16 kartų greičiau nei kros-koreliacijos ir 4-8 kartus greičiau nei suderinto filtro metodas, kai transporto priemonės greitis yra intervale nuo 35,8 km/h iki 65,2 km/h, o matavimo santykinė paklaida neviršija 2,8 % duotame greičių diapazone.

INFORMACIJOS ŠALTINIŲ SĄRAŠAS

1. Osman Ibrahim, Hazem ElGendy, Ahmed N. SIshafee, „Speed Detection Camera System using Image Processing Techniques on Video Streams“, International Journal of Computer and Electrical Engineering, Vol. 3, No. 6, 2011.
2. Adnan, M.A., Zainuddin, N.I., Sulaiman, N., Besar, T.B.H.T., Vehicle Speed Measurement Technique Using Various Speed Detection Instrumentation, 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC 2013), 668-672, 2013.
3. Karla Panchuk, Physical Geology, Second Adapted Edition, Par. 3.4.
4. Martin Isaksson, „Vehicle Detection using Anisotropic Magnetoresistors“, Report no. EX034/2008, Department of Signals And Systems, Chalmers University of Technology, 2007.
5. Michael J. Caruso, Lucky S. Withanawasam, „Vehicle Detection and Compass applications using AMR Magnetic Sensors“, Honeywell, SSEC, 12001 State Highway 55, Plymouth, MN USA 55441
6. Hongmei Zhu, Fengqi Yu, „A Cross-Correlation Technique for Vehicle Detections in Wireless Magnetic Sensor Network“, IEEE Sensors Journal, Vol. 16, NO. 11, 2016.
7. Xu Bao, Haijian Li, Dongwei Xu, Limin Jia, Bin Ran, Jian Rong, „Traffic Vehicle Counting in Jam Flow Conditions Using Low-Cost and Energy-Efficient Wireless Magnetic Sensors“, Sensors Journal, 2016.
8. H. Zhu and F. Yu, “A vehicle parking detection method based on correlation of magnetic signals,” Int. J. Distrib. Sensor Netw., p. 101, 2015.
9. Y. Iwasaki, “A method of robust moving vehicle detection for bad weather using an infrared thermography camera,” in Proc. IEEE Conf. Wavelet Anal. Pattern Recognit. (ICWAPR), Aug. 2008, pp. 86–90.
10. Ramón V. Cirilo Gimeno, Marta Pla Castells, Antonio García Celda and Javier Martínez Plumé, „A Simulation model using AMR sensors applied to vehicles classification and identification“, University of Valencia, E 46980, Paterna, Spain, May, 2013.
11. Peter Šarčević, „Issues and challenges of of intelligent systems and computational intelligence“, Studies in computational intelligence 530, DOI: 10.1007/978-3-319-03206-1_8, Switzerland, 2014.
12. Djamel Djenouri, Messaoud Doudou, Mohamed Amine Kafi, „An Oscillation-based Algorithm for Reliable Vehicle Detection with Magnetic Sensors“, The 2nd International conference on control, automation and robotics. Electronic ISBN: 978-1-4673-9859-6, 2016.

13. Jianping Wu, Zhaobin Liu, Jinxiang Li, Caidong Gu, Maoxin Si, Fangyong Tan, "An Algorithm for Automatic Vehicle Speed Detection using Video Camera", JiangSu Province Support Software Engineering R&D Center for Modern Information Technology Application in Enterprise, Suzhou, China, 215104.
14. Yiyang Wang, Yuexian Zou, Hang SHI, He ZHAO, „Video Image Vehicle Detection System for Signaled traffic Intersection“, Ninth International Conference on Hybrid Intelligent Systems, 2009.
15. Kenji Kodera, Akitoshi Itai, Hiroshi Yasukawa, „Approaching Vehicle Detection Using Linear Microphone Array“, International Symposium on Information Theory and its Applications, New Zeland, 2008.
16. A. Severdaks, M. Liepins, „Vehicle Counting and Motion Direction Detection Using Microphone Array“, ELEKTRONIKA IR ELEKTROTECHNIKA, ISSN 1392-1215, VOL. 19, NO. 8, 2013.
17. Youngtae Jo, Jinsup Choi, and Inbum Jung, „Traffic Information Acquisition System with Ultrasonic Sensors in Wireless Sensor Networks“, International Journal of Distributed Sensor Networks Volume 2014, Article ID 961073, 2014.
18. S. Sheik Mohammed Ali, Bobby George, Lelitha Vanajakshi, and Jayashankar Venkatraman. „A Multiple Inductive Loop Vehicle Detection System for Heterogeneous and Lane-Less Traffic“, IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, VOL. 61, NO. 5, MAY 2012.
19. Zbigniew Marszalek, Ryszard Sroka, Tadeusz Zeglen, „Inductive Loop for Vehicle Axle Detection from First Concepts to the System Based on Changes in the Sensor Impedance Components“, Electronic ISBN: 978-1-4799-8701-6, 2015.
20. V. Markevičius, D. Navikas, M. Žilys, D. Andriukaitis, A. Valinevičius, M. Čepėnas, „Dynamic Vehicle Detection via the use of Magnetic Field Sensors“. Sensors. Basel: MDPI AG. ISSN 1424-8220. 2016, vol. 16, iss. 1, Article 78, p. [1–9].
21. ShangGuan Wei, Wang Jian, Cai Bai-gen, Yin Qin, „An Novel Vehicle Detection Method Based on Wireless Magneto-resistive Sensor“, Third International Symposium on Intelligent Information Technology Application, DOI: , 2009.
22. V. Markevičius, D. Navikas, M. Žilys, D. Andriukaitis, A. Valinevičius, A. Idzkowski, „Vehicle Speed and Length Estimation Using Data from Two Anisotropic Magneto-Resistive (AMR) Sensors“. Sensors. Basel: MDPI AG. ISSN 1424-8220. 2017, vol. 17, iss. 8, Article 1778, p. [1–13].

PRIEDAI

Priedas 1. MATLAB programa

```
close all;
clear all;
% System parameters whihc were used
fd = 2e3; % Sampling frequency is 2 kHz
Td = 1/fd; % discretization period
% Atskaitru indeksai, kurie naudojami signalu iskirpimui
start_index = 800;
end_index = 2000;
% start_index = 1160;
% end_index = 1700;
% Load signals
data = csvread('015 2017-11-27_10-10-17-272.csv');
realData = csvread('realDuomenys.csv');
% Get real signals with real speeds
rSig1 = realData(:,8);
rSig2 = realData(:,9);
rSig1z = realData(:,4);
rSig2z = realData(:,7);
speed1Volvo1sig = round(transpose(rSig1(1:800,1)));
speed1Volvo2sig = round(transpose(rSig2(1:800,1)));
speed1Volvo1sigZ = round(transpose(rSig1z(1:800,1))); % Pirmas greitis v = 35,8
speed1Volvo2sigZ = round(transpose(rSig2z(1:800,1))); % Pirmas greitis v = 35,8
% speed1Volvo1sig = round(transpose(rSig1(1300:1800,1)));
% speed1Volvo2sig = round(transpose(rSig2(1300:1800,1)));
speed2Volvo1sig = round(transpose(rSig1(2500:3000,1)));
speed2Volvo2sig = round(transpose(rSig2(2500:3000,1)));
speed2Volvo1sigZ = round(transpose(rSig1z(2500:3000,1))); % Antras greitis v = 50,4
speed2Volvo2sigZ = round(transpose(rSig2z(2500:3000,1))); % Antras greitis v = 50,4
% speed2Volvo1sig = round(transpose(rSig1(3750:4050,1)));
% speed2Volvo2sig = round(transpose(rSig2(3750:4050,1)));
speed3Volvo1sig = round(transpose(rSig1(4250:4550-1,1)));
speed3Volvo2sig = round(transpose(rSig2(4250:4550-1,1)));
speed3Volvo1sigZ = round(transpose(rSig1z(4250:4550,1))); % Trecias greitis v = 86,5
speed3Volvo2sigZ = round(transpose(rSig2z(4250:4550,1))); % Trecias greitis v = 86,5
% speed3Volvo1sig = round(transpose(rSig1(5450:5750,1))); % Trecias greitis v = 86,5
% speed3Volvo2sig = round(transpose(rSig2(5450:5750,1))); % Trecias greitis v = 86,5

% wSize = 10;
% speed1Volvo1sig = customMovingAverage(speed1Volvo1sig, wSize);
% speed1Volvo2sig = customMovingAverage(speed1Volvo2sig, wSize);
% speed2Volvo1sig = customMovingAverage(speed2Volvo1sig, wSize);
% speed2Volvo2sig = customMovingAverage(speed2Volvo2sig, wSize);
% speed3Volvo1sig = customMovingAverage(speed3Volvo1sig, wSize);
% speed3Volvo2sig = customMovingAverage(speed3Volvo2sig, wSize);

% Load modules of both signals
sig1 = data(:,4);
sig2 = data(:,8);
%Load z axis of both signals
sig1z = data(:,3);
sig2z = data(:,7);
%Load y axis of both signals
sig1y = data(:,2);
sig2y = data(:,6);
%Load x axis of both signals
sig1x = data(:,1);
sig2x = data(:,5);
%Transpose all signal components and modules and take only part of data for signatures
sig1=transpose(sig1(start_index:end_index-1,1));
sig2=transpose(sig2(start_index:end_index-1,1));
sig1z=transpose(sig1z(start_index:end_index-1,1));
sig2z=transpose(sig2z(start_index:end_index-1,1));
sig1y=transpose(sig1y(start_index:end_index-1,1));
sig2y=transpose(sig2y(start_index:end_index-1,1));
sig1x=transpose(sig1x(start_index:end_index-1,1));
sig2x=transpose(sig2x(start_index:end_index-1,1));

% sig1 = speed3Volvo1sig;
% sig2 = speed3Volvo2sig;

% sImax = max(abs(sig1));
```

```

% s2max = max(abs(sig2));
% sig1 = sig1./s1max;
% sig2 = sig2./s2max;

% wSize = 20;
% sig1 = customMovingAverage(sig1, wSize);
% sig2 = customMovingAverage(sig2, wSize);
% Pakeisti kintamuosius i tokius, kad sig1 ir sig2 laikytu z asies reiksmes
% sig1 = sig1z;
% sig2 = sig2z;
% Metodai ir ju testavimas
% 1 ----- Kros-koreliacija
tic
[r, lags] = xcorr(sig1, sig2); % Calculate crosscorrelation function
[rmax, maxIndex] = max(abs(r)); % get index of maximum element
result_method1 = abs(length(sig1) - maxIndex); % Calculate lagg between samples
time_method1 = toc;
% -----
% 2 ----- Masiu centro metodas
tic
mas1_sum = 0;
mas2_sum = 0;
mas1_N = 0;
mas2_N = 0;
for a = 1:length(sig1)
    mas1_sum = mas1_sum + sig1(a);
    mas2_sum = mas2_sum + sig2(a);

    mas1_N = mas1_N + (sig1(a)*a);
    mas2_N = mas2_N + (sig2(a)*a);
end
n1 = mas1_N/mas1_sum;
n2 = mas2_N/mas2_sum;
result_method2 = abs(n2-n1);
time_method2 = toc;
% -----
% 3 ----- Atstumo tarp maximumu metodas
tic
[~, max1idx] = max(sig1);
[~, max2idx] = max(sig2);
result_method3 = abs(max2idx-max1idx);
time_method3 = toc;
% -----
% 4 ----- Slenkstines vertes virsijimo metodas
tic
Threshold = 250; % Slenkstine verte nustatoma empiriskai
index1 = 0;
index2 = 0;
for a = 1:length(sig1)
    if sig1(a) > Threshold
        index1 = a;
        break;
    end
end
for a = 1:length(sig1)
    if sig2(a) > Threshold
        index2 = a;
        break;
    end
end
result_method4 = abs(index2 - index1);
time_method4 = toc;
% -----
% Pries testuojant šit? metod?, reikia pakeisti virsuje esancius
% kintamuosius:
% start_index = 1160;
% end_index = 1700;
% 5 ----- Slenkstines vertes virsijimo metodas (slenkstis lygus pirmo
% signalo vidurkiui
tic
Threshold = mean(sig1); % Slenkstine verte nustatoma empiriskai
ThresholdMean = Threshold;
index1 = 0;
index2 = 0;
for a = 1:length(sig1)
    if sig1(a) > Threshold

```

```

        index1 = a;
        break;
    end
end
for a = 1:length(sig1)
    if sig2(a) > Threshold
        index2 = a;
        break;
    end
end
result_method5 = abs(index2 - index1);
time_method5 = toc;
% -----
% 6 ---- Figuros apribotos slenkstinemis vertemis is abieju pusių vidurkio
% metodas
tic
Threshold = 300;
min_distance_from_first_treshold = 100;
up1idx = 0;
up2idx = 0;
down1idx = 0;
down2idx = 0;
for a = 1:length(sig1)
    if sig1(a) > Threshold
        up1idx = a;
        break;
    end
end
for a = 1:length(sig1)
    if sig2(a) > Threshold
        up2idx = a;
        break;
    end
end
for a = up1idx+min_distance_from_first_treshold:length(sig1)
    if sig1(a) < Threshold
        down1idx = a;
        break;
    end
end
for a = up2idx+min_distance_from_first_treshold:length(sig1)
    if sig2(a) < Threshold
        down2idx = a;
        break;
    end
end
result_method6 = abs((down1idx-down2idx) + (up1idx-up2idx))/2;
time_method6 = toc;
% -----
% 7 ---- Figuros apribotos slenkstinemis vertemis centro metodas
% (slenkstis parenkamas empiriskai), metodui reikalinga ivesti parametra,
% kuris nustato koks minimalus atstumas turi buti nuo pakilimo virs
% slenkstines ribos, iki nukritimo zemiau slenkstines ribos
tic
Threshold = 200;
min_distance_from_first_treshold = 400;
up1idx = 0;
up2idx = 0;
down1idx = 0;
down2idx = 0;
for a = 1:length(sig1)
    if sig1(a) > Threshold
        up1idx = a;
        break;
    end
end
for a = 1:length(sig1)
    if sig2(a) > Threshold
        up2idx = a;
        break;
    end
end
for a = up1idx+min_distance_from_first_treshold:length(sig1)
    if sig1(a) < Threshold
        down1idx = a;
        break;
    end
end

```

```

end
for a = up2idx+min_distance_from_first_treshold:length(sig1)
    if sig2(a) < Treshold
        down2idx = a;
        break;
    end
end
result_method7 = abs((down1idx-up2idx)/2 - (down2idx-up1idx)/2);
time_method7 = toc;
% -----
% 8----- Maziausiu kvadratu skirtumu metodus
tic
ssd = 0;
skirt = zeros(1, length(sig1));
diff = zeros(1, length(sig1));
diff1 = zeros(1, length(sig1));
sse1 = 0;
sse2 = 0;
delta = 0;

for a = 1:length(sig1)
    delta = 0;
    for b = 1: length(sig2)
        k = b + a;
        if k < length(sig2)
            delta = delta + power(abs(sig2(b)-sig1(k)),2);
        else
            break;
        end
    end
    diff(a) = delta;
end
[~,result_method8] = min(diff(1:length(sig1)/2));
time_method8 = toc;
% -----
% 9----- Maziausiu skirtumo metodus
tic
for a = 1:length(sig1)
    delta = 0;
    for b = 1: length(sig2)
        k = b + a - 1;
        if k < length(sig2) + 1
            delta = delta + abs(sig2(b)-sig1(k));
        else
            break;
        end
    end
    diff1(a) = delta;
end
[~,result_method9] = min(diff1(1:length(sig1)/2));
time_method9 = toc;
% -----
% 10----- Vienodu plotu metodus
% tic
% Treshold = 300;
% min_distance_from_first_treshold = 100;
% for a = 1:length(sig1)
%     if sig1(a) > Treshold
%         index1 = a;
%         break;
%     end
% end
% for a = 1:length(sig2)
%     if sig2(a) > Treshold
%         index2 = a;
%         break;
%     end
% end
% for a = index1+min_distance_from_first_treshold:length(sig1)
%     if sig1(a) < Treshold
%         index1_e = a;
%         break;
%     end
% end
% for a = index2+min_distance_from_first_treshold:length(sig2)
%     if sig2(a) < Treshold
%         index2_e = a;

```

```

% break;
% end
% end
% c1 = round((index1_e-index1)/2) + index1;
% c2 = round((index2_e-index2)/2) + index2;
% %Skaiciuojam sumas abiems signalams nuo tresholdu kirtimo iki centro
% area1a = 0;
% area2a = 0;
% area1b = 0;
% area2b = 0;
% for a = index1:c1
%   area1a = area1a + sig1(a);
% end
% for a = index2:c2
%   area2a = area2a + sig2(a);
% end
% for a = c1:index1_e
%   area1b = area1b + sig1(a);
% end
% for a = c2:index2_e
%   area2b = area2b + sig2(a);
% end
% % lyginam plotus kiekvienai daliai
% if area1a > area1b
%   pp = 1;
%   while area1a > area1b
%     area1a = area1a - sig1(c1 - pp);
%     area1b = area1b + sig1(c1 + pp);
%     pp = pp + 1;
%   end
%   centre1 = index1 + c1 - pp + 2;
% else
%   pp = 1;
%   while area1a < area1b
%     area1a = area1a + sig1(c1 + pp);
%     area1b = area1b - sig1(c1 - pp);
%     pp = pp + 1;
%   end
%   centre1 = index1 + c1 - pp + 2;
% end
% if area2a > area2b
%   pp = 1;
%   while area2a > area2b
%     area2a = area2a - sig2(c2 - pp);
%     area2b = area2b + sig2(c2 + pp);
%     pp = pp + 1;
%   end
%   centre2 = index2 + c2 - pp + 2;
% else
%   pp = 1;
%   while area1a < area1b
%     area2a = area2a + sig2(c2 + pp);
%     area2b = area2b - sig2(c2 - pp);
%     pp = pp + 1;
%   end
%   centre2 = index2 + c2 - pp + 2;
% end
%
% result_method10 = abs(centre2-centre1);
% time_method10 = toc;
% -----
% 11----- Suderinto filtro metodas (Cross correlation dazniu srity)
tic
L = length(sig1');
N = length(sig2');
matched_filter = flip(sig2');
matched_filter_Res = fft(matched_filter, N);
corr_fft = matched_filter_Res.*fft(sig1');
corr_out = abs(iff(corr_fft));
[M, result_method11] = max(corr_out);
fft_lag_magnitude=abs(result_method11);
% matched_filter = flip(sig2');
% matched_filter_Res = fft(matched_filter, 512);
% corr_fft = matched_filter_Res.*fft(sig1, 512);
% corr_out = abs(iff(corr_fft));
% [M, result_method11] = max(corr_out);
% fft_lag_magnitude=abs(result_method11);

```



```

time_method11 = toc;
% -----

% wSize = 10;
% sig1 = customMovingAverage(sig1, wSize);
% sig2 = customMovingAverage(sig2, wSize);
% -----
% 11----- Dalines kros-koreliacijos metodas
% Visuose rezultatas gaunasi 21 del to, kad kros-koreliacijos funkcijos
% virsune yra kampuota, paziureti kros-koreliacijos virsune. Pries
% paduodant signalus tokiam apdorojimui reiketu suvidurkinti signalus
tic
result_method12a = find_lagg(sig1, sig2, length(sig1), uint32(result_method2)); % pagalbinis: masiu centras
toc
result_method12b = find_lagg(sig1, sig2, length(sig1), result_method3); % pagalbinis: Atstumas tarp maximumu
result_method12c = find_lagg(sig1, sig2, length(sig1), result_method4); % pagalbinis: slenkstines reiksmes virsijimas
result_method12d = find_lagg(sig1, sig2, length(sig1), result_method5); % pagalbinis: slenkstines reiksmes virsijimas (slenkstis lygus signalo
vidurkiui)
result_method12e = find_lagg(sig1, sig2, length(sig1), uint32(result_method6)); % pagalbinis: 6 metodas
result_method12f = find_lagg(sig1, sig2, length(sig1), uint32(result_method7)); % pagalbinis: 7 metodas
% result_method12g = find_lagg(sig1, sig2, length(sig1), uint32(result_method10)); % pagalbinis: 7 metodas
% -----

% 11----- Ivestines metodas
% wSize = 50;
% sig1 = customMovingAverage(sig1, wSize);
% sig2 = customMovingAverage(sig2, wSize);
Treshold = 500;
tr_index = 0;
for a = 1:length(sig1)-1
    if sig1(a) < Treshold
        tr_index = a;
        break;
    end
end
for a = 1:length(sig1)-1
    deriv1(a) = (sig1(a+1)-sig1(a))/Td;
    deriv2(a) = (sig2(a+1)-sig2(a))/Td;
end
% -----
%
% % Originalus pirmo signalo vaizdas: x,y ir z asys bei modulis.
% figure(1)
% hold on
% plot(sig1x,'--');
% plot(sig1y,'--');
% plot(sig1z,'--');
% plot(sig1);
% hold off
% legend('x1', 'y1', 'z1', 'modulis1');
% xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
% ylabel('Magnetin{\.e} indukcija [uT]', 'Interpreter','Latex');
% grid on;
% Abieju signalu moduliai.
figure(2)
hold on
plot(sig1, 'Color', 'b');
plot(sig2, 'Color', 'k');
plot(300*ones(length(sig1)), 'Color', 'r');
hold off
legend('modulis2', 'modulis1', 'Slenkstis');
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Magnetin{\.e} indukcija [uT]', 'Interpreter','Latex');
grid on;
% % Maziausiu skirtumo ir maziausiu skirtumo kvadrato
%
figure(3)
subplot(1,2,1);
hold on
plot(diff, 'Color', 'b');
plot(diff1, 'Color', 'k');
xlim([0 length(sig1)/2]);
legend('MKS', 'MS');

```

```

xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Amplitud{\.e}', 'Interpreter','Latex');
grid on;
hold off
subplot(1,2,2);
hold on
plot(diff, 'Color', 'b');
plot(diff1, 'Color', 'k');
legend('MKS', 'MS');
xlim([10 30]);
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Amplitud{\.e}', 'Interpreter','Latex');
grid on;
hold off
grid on;
% Metodo 11 su matched filter rezultatas kros koreliacine funkcija laiko
% % asyje
figure(4)
subplot(1,2,1);
plot(corr_out, 'Color', 'k');
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Amplitud{\.e}', 'Interpreter','Latex');
xlim([0 length(corr_out)/2]);
grid on
subplot(1,2,2);
plot(corr_out, 'Color', 'k');
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Amplitud{\.e}', 'Interpreter','Latex');
xlim([15 25]);
grid on;
%
%
% Kroskoreliacijos metodus.
figure(5)
subplot(1,2,1);
plot(lags, r, 'Color', 'k');
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Amplitud{\.e}', 'Interpreter','Latex');
xlim([-1000 1000]);
grid on;
subplot(1,2,2);
plot(lags, r, 'Color', 'k');
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
ylabel('Amplitud{\.e}', 'Interpreter','Latex');
xlim([15 25]);
grid on;

% % 6 Metodo signalai su tresholdu.
% % figure(6)
% % hold on
% % plot(sig1, 'Color', 'b');
% % plot(sig2, 'Color', 'k');
% % plot(TresholdMean*ones(length(sig1)), 'Color', 'r');
% % xlim([0 length(sig1)]);
% % xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
% % ylabel('Magnetin{\.e}} indukcija [uT]', 'Interpreter','Latex');
% % hold off
% % legend('Modulis1', 'Modulis2', 'Slenkstis');
% % grid on;
%
% % figure(7)
% % hold on
% % plot(sig1, 'Color', 'b');
% % plot(sig2, 'Color', 'k');
% % plot(300*ones(length(sig1)), 'Color', 'r');
% % xlim([0 length(sig1)]);
% % xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
% % ylabel('Magnetin{\.e}} indukcija [uT]', 'Interpreter','Latex');
% % hold off
% % legend('Modulis1', 'Modulis2', 'Slenkstis');
% % grid on;
%
% % Signalu interpoliavimas ir rezultatu patikslinimas
% % Interpoliuoti signalus n kartu
% n = 10;
% nsig = length(sig1);
% naxis = 1:length(sig1);

```

```

% naxis_interp = 1:1/n:length(sig1);
% sig1_interp = interp1(naxis,sig1,naxis_interp);
% sig2_interp = interp1(naxis,sig2,naxis_interp);
%
% [rinterp_sig, lags_interp] = xcorr(sig1_interp, sig2_interp); % Calculate crosscorrelation function
% [maxValueinter, maxIndexinterp] = max(abs(rinterp_sig)); % get index of maximum element
% lagDiff_interp = length(lags_interp)-maxIndexinterp; % Calculate lagg between samples
%
% rinterp_sig = rinterp_sig./maxValueinter;
% r = r./rmax;
%
% nr1 = length(r);
% % nNew = lags(1):0.1:length(sig1)-1;
% nNew = lags_interp(1):10:length(sig1)*10-10;
%
% % figure(12)
% % hold on
% % plot(lags, r);
% % plot(nNew,r,'-o');
% % stem(lags_interp,rinterp_sig,'-o');
% % hold off
% % ylim([0.995 1.001]);
% % xlim([180 220]);
% xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
% ylabel('Normuota Amplitud{\.e}', 'Interpreter','Latex');
% legend('Neinterpoliuota', 'Interpoliuota');
% grid on;
%
% % figure(13)
% % hold on
% % plot(deriv1);
% % plot(deriv2);
% % hold off
% % ylim([0.995 1.001]);
% % xlim([180 220]);
% % xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
% % ylabel('Normuota Amplitud{\.e}', 'Interpreter','Latex');
% % legend('Neinterpoliuota', 'Interpoliuota');
% grid on;
%
% % Slenkanciu vidurkiu itaka
% wSize = 10;
% sig1avg = customMovingAverage(sig1, wSize);
% sig2avg = customMovingAverage(sig2, wSize);
% sig1xavg = customMovingAverage(sig1x, wSize);
% sig2xavg = customMovingAverage(sig2x, wSize);
% sig1yavg = customMovingAverage(sig1y, wSize);
% sig2yavg = customMovingAverage(sig2y, wSize);
% sig1zavg = customMovingAverage(sig1z, wSize);
% sig2zavg = customMovingAverage(sig2z, wSize);
%
% figure(15);
% subplot(2,1,1);
% hold on
% title('Pradiniai signalai', 'Interpreter','Latex')
% plot(sig1x,'--');
% plot(sig1y,'--');
% plot(sig1z,'--');
% plot(sig1);
% ylim([-1500 2500]);
% hold off
% legend('x1', 'y1', 'z1', 'modulis1');
% xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');
% ylabel('Magnetin{\.e} indukcija [uT]', 'Interpreter','Latex');
% grid on
% subplot(2,1,2);
% hold on
% title('Signalai po slenkan{\v{c}io} vidurkio filtro', 'Interpreter','Latex')
% plot(sig1xavg,'--');
% plot(sig1yavg,'--');
% plot(sig1zavg,'--');
% plot(sig1avg);
% ylim([-1500 2500]);
% xlim([0 length(sig1xavg)-wSize]);
% hold off
% legend('x1', 'y1', 'z1', 'modulis1');
% xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter','Latex');

```

```

% ylabel('Magnetin{\e} indukcija [uT]', 'Interpreter', 'Latex');
% grid on;
%
% wSize = 10;
% sig1avg1 = customMovingAverage(sig1, 10);
% sig1avg2 = customMovingAverage(sig1, 20);
% sig1avg3 = customMovingAverage(sig1, 40);
% sig1avg4 = customMovingAverage(sig1, 60);
% sig1avg5 = customMovingAverage(sig1, 80);
%
% figure(16);
% hold on
% title('Pradiniai signalai', 'Interpreter', 'Latex')
% plot(sig1);
% plot(sig1avg1);
% plot(sig1avg2);
% plot(sig1avg3);
% plot(sig1avg4);
% plot(sig1avg5);
% ylim([0 2500]);
% hold off
% legend('Lango dydis = 0', 'Lango dydis = 10', 'Lango dydis = 20',...
%       'Lango dydis = 40', 'Lango dydis = 60', 'Lango dydis = 80');
% xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter', 'Latex');
% ylabel('Magnetin{\e} indukcija [uT]', 'Interpreter', 'Latex');
% grid on;
%
% %Real volvo signals
% figure(17);
% hold on
% plot(rSig1);
% plot(rSig2);
% hold off
% xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter', 'Latex');
% ylabel('Magnetin{\e} indukcija [uT]', 'Interpreter', 'Latex');
% grid on;
%
% %Real volvo signals
figure(18);
subplot(3,1,1);
hold on
plot(speed1Volvo1sig, 'Color', 'b');
plot(speed1Volvo2sig, 'Color', 'k');
hold off
xlim([100 700]);
legend('v = 35,8 km/h')
grid on;
subplot(3,1,2);
hold on
plot(speed2Volvo1sig, 'Color', 'b');
plot(speed2Volvo2sig, 'Color', 'k');
hold off
xlim([50 450]);
legend('v = 50,4 km/h')
ylabel('Magnetin{\e} indukcija [uT]', 'Interpreter', 'Latex');
grid on;
subplot(3,1,3);
hold on
plot(speed3Volvo1sig, 'Color', 'b');
plot(speed3Volvo2sig, 'Color', 'k');
hold off
xlim([0 300]);
legend('v = 65,2 km/h')
xlabel('Atskait{\c{u}} skai{\v{c}ius} N', 'Interpreter', 'Latex');
grid on;

% Skaiciuojam kros-koreliacija ir greiti visais trim atvejais
[rV1, lagsV1] = xcorr(speed1Volvo1sig, speed1Volvo2sig); % Calculate crosscorrelation function
[rmax, maxIndex] = max(abs(rV1)); % get index of maximum element
rm1 = abs(length(speed1Volvo1sig) - maxIndex); % Calculate lagg between samples

v1 = (0.3/((1/fd)*rm1))*3600/1000;

[rV2, lagsV2] = xcorr(speed2Volvo1sig, speed2Volvo2sig); % Calculate crosscorrelation function
[rmax, maxIndex] = max(abs(rV2)); % get index of maximum element
rm2 = abs(length(speed2Volvo1sig) - maxIndex); % Calculate lagg between samples

```

```

v2 = (0.3/((1/fd)*rm2))*3600/1000;

[rv3, lagsv3] = xcorr(speed3Volvo1sig, speed3Volvo2sig); % Calculate crosscorrelation function
[rmax, maxIndex] = max(abs(rv3)); % get index of maximum element
rm3 = abs(length(speed3Volvo1sig) - maxIndex); % Calculate lagg between samples

v3 = (0.3/((1/fd)*rm3))*3600/1000;

% Z komponent?s 2.16 skyriui apie aptartu velinimo tarp dviej? signalu
% skaiciavimo budu pritaikyma z asiai.
%
figure(19);
hold on
plot(sig1z, 'Color', 'b');
plot(sig2z, 'Color', 'k');
hold off
legend('signalas2 z', 'signalas1 z');
xlabel('Atskait{\c{u}} skai{\v{c}}ius N', 'Interpreter', 'Latex');
ylabel('Magnetin{\e} indukcija [uT]', 'Interpreter', 'Latex');
grid on;

vm1 = (0.3/((1/fd)*result_method1))*3600/1000;
vm2 = (0.3/((1/fd)*result_method2))*3600/1000;
vm3 = (0.3/((1/fd)*result_method3))*3600/1000;
vm4 = (0.3/((1/fd)*result_method4))*3600/1000;
vm5 = (0.3/((1/fd)*result_method5))*3600/1000;
vm6 = (0.3/((1/fd)*result_method6))*3600/1000;
vm7 = (0.3/((1/fd)*result_method7))*3600/1000;
vm8 = (0.3/((1/fd)*result_method8))*3600/1000;
vm9 = (0.3/((1/fd)*result_method9))*3600/1000;
% vm10 = (0.3/((1/fd)*result_method10))*3600/1000;
vm11 = (0.3/((1/fd)*result_method11))*3600/1000;
%
% figure(20);
% hold on
% plot(speed1Volvo1sigZ);
% plot(speed1Volvo2sigZ);
% hold off
% grid on;

```